Wm. Dudley
9/15/66

S-C 4020 MICROFILM RECORDER

USER'S MANUAL

~~M. Fleming~~ M. K. Reilley
3/29/67

MEMORANDUM

March 24, 1967

TO: S-C 4020 Programmers

FROM: Grant Carrington

SUBJECT: Corrections to S-C 4020 Manual

The following corrections and additions should be made to the S-C 4020 Microfilm Recorder User's Manual, prepared by CSC for GSFC under Contract No. NAS5-9758-39:

Page II-11: In the description of the DX,DY arguments for GRID1V the following should be added: "However, a GRID1V error mark will be displayed. This mark cannot be suppressed."

Page II-80: The calling sequence for DSHLNV should be changed from

```
CALL DSHINV (IX1, IY1, IX2, IY2)
                to
CALL DSHINV (IX1, IY1, IX2, IY2, IL, IS)
```

The IL, IS arguments are the same as for INCRV. There is no INCRV for the Lockheed package.

Page II-119: The third sentence on this page should be changed from "Using SCORS IV and the GSFC IBSYS system (WRITE (7, fmt)..." to "Using SCORS IV and the GSFC IBSYS system WRITE (61,fmt)..."

Page A-19: The routine name "NIPT$" should be "NINPT$"

Appendix B: Two more routines have been written for use with the SCORS and Lockheed packages, by Barbara Walton of the Information Processing Division. These routines have been written in FORTRAN IV and were developed on the Univac 1108, but they should be readily adaptable to the IBM 7094 SCORS packages.

CIRCLE draws a circle of given radius and center.

```
CALL CIRCLE (NX, NY, NR)
```

where

NX = x-raster position of center
NY = y-raster position of center
NR = radius of circle in rasters

Subj: Corrections to S-C 4020 Manual

If the circle goes beyond the frame, only that portion of the circle within the frame will be drawn.

POLAR may be used to make a polar stereographic projection of two arrays. All points are projected on the equatorial plane as viewed from the north.

CALL POLAR (NP, GLAT, GLONG)

where

NP = number of points to be plotted from GLAT, GLONG arrays
GLAT = latitude array
GLONG = longitude array.

Latitude and longitude must be given in relation to an ideal sphere, e.g., as in celestial or galactic coordinate systems.

Copies of these routines are available from the Program Librarian.

Grant Carrington

Grant Carrington
Programming Methods Section

S-C 4020 MICROFILM RECORDER

USER'S MANUAL

Contract No.: NAS5-9758-39

Prepared by

COMPUTER SCIENCES CORPORATION
8121 Georgia Avenue
Silver Spring, Maryland 20910

for

Goddard Space Flight Center
Greenbelt, Maryland 20771

TABLE OF CONTENTS

Page

Page

INTRODUCTION

INTRODUCTION

Several major plotting packages for the S-C 4020 Microfilm Recorder (also known as a Plotter) are available at the Goddard Space Flight Center (GSFC). These packages, implemented for use on the UNIVAC 1107/1108 or the IBM 7094, may be summarized as follows:

A. SCORS II

SCORS II, written for FORTRAN II compilations on the IBM 7094, is the original North American Aviation (NAA) Plot Utility Package for the S-C 4020. Stromberg Carlson, having assumed responsibility for its maintenance and distribution (via UAIDE), called this package SCORS and later, to distinguish it from the FORTRAN IV version, renamed it SCORS II.

B. SCORS IV

SCORS IV, which is a direct derivation of SCORS II, is FORTRAN IV oriented and also implemented for the IBM 7094.

C. LOCKHEED PACKAGE

The Lockheed Package is also modeled after the original NAA package; but it responds to UNIVAC 1107/1108 FORTRAN operating under the EXEC II monitoring system. Again, Stromberg Carlson has assumed maintenance and distribution responsibility.

D. CSC PLOTTER PROGRAMMING SYSTEM

The CSC System, which has also been implemented for use on the UNIVAC 1107/1108, is unlike the above packages in design. It is a fundamental plotting system and is coded quite differently than the systems mentioned previously.

Although all four packages are plotting systems for the S-C 4020, each is described in a separate document. As each package was initially implemented, slight alterations were made, which caused difficulties in the task of updating the various documents.

In addition, other routines were beginning to be developed and their descriptions were not readily available to interested programmers. Furthermore, it became apparent that additional routines and packages - such as the one for the IBM 360 system - would come into existence.

As a result, it was decided to collect some of the documentation now existing and to incorporate it into a single volume - inserting the package modifications made so far. Such a volume could be expanded with respect to future changes or insertions with far greater ease.

This manual is the result of the previous decision. It introduces the programmer to the S-C 4020 hardware, serves as a reference document to the various plotter packages and routines currently being used at the GSFC, and instructs in the operational procedures which exist at the Center. Now - when a potential S-C 4020 user seeks information, a single, up-to-date document can be furnished.

ORGANIZATION OF DOCUMENT

The S-C 4020 hardware is described in Section I. The new user is especially urged to read this section before proceeding to the following sections in order to become familiar with the equipment, its applications and its terminology.

The three similar packages--SCORS II, SCORS IV, and LOCKHEED--are presented as a single system in Section II with each package's exceptions noted. The CSC System, since it was not patterned after the North American Aviation package (SCORS II), is discussed independently in Section III.

At first glance, the list of S-C 4020 routines of Section II seems formidable indeed, but a user should not feel baffled at the great amount of details available. Some routines were developed for special purposes with limited usage, while many are used as lower-level modules for more general subroutines - rarely used directly by the programmer.

This may prompt the new user to ask: what routines do I need to know? In answer, the following list was prepared and may be referenced through the Table of Contents:

Insertion of ID Information: IDFRMV
Selecting the Camera: CAMRAV
Advancing the Film: FRAMEV, RESETV
Frame Count Retrieval: NOFRV
Generating a Grid: GRID1V
Computation of GRID1V Arguments: DXDYV
Intensity Selection: BRITEV, FAINTV
Plotting an Array: APLOTV
Plotting Individual Data Points: POINTV
Basic Plot-Print Subprogram: PLOTV
Typewriter Printing: PRINTV
Vertical Titles: APRNTV
Display of Non-FORTRAN Characters

Fixed Point Labels: LABLV
Conversion: NXV, NYV
Line Generation: LINEV
Axis Lines: XAXISV, YAXISV

The CSC System, in Section III, is presented in two parts - Utility and Basic Subroutines. Here, the Utility Subroutines PLBGIN and PLOT should be of prime concern to the new user of this system.

Section IV discusses the operational procedures for the available computers at the GSFC, including the forms to be completed in order to operate within a system. Finally, Section V briefly presents various programmable operations which may give the plotter programmer an insight as to what can be accomplished on the S-C 4020 Microfilm Recorder.

In concluding this introduction, it should be mentioned that descriptions of newly developed routines - or of those routines not included within this document - should be forwarded to the Programming Methods Section located in Building No. 3, Room 125 at the GSFC. There the material can be reviewed and distributed for incorporation into this document.

Since this document contains a considerable amount of edited composition from documentation describing the systems in existence, credit is hereby extended to R. Proper, author at North American Aviation, and the other interested personnel - both at Goddard and the Computer Sciences Corporation - who contributed.

SECTION I

THE S-C 4020 MICROFILM RECORDER

THE S-C 4020 MICROFILM RECORDER

The S-C Microfilm Recorder, designed and manufactured by General Dynamics/ Electronics, San Diego, California, operates as a peripheral plotting and recording device for the IBM 7094 and the UNIVAC 1107/1108 high-speed digital computers at the GSFC. This section describes the features of the S-C 4020 and gives the basic programming requirements including the coding and word format of commands used for control and recording operations.

The S-C 4020 is a peripheral device which reads magnetic tape (output from a computer program) and produces graphical, pictorial or alphanumeric output. As the tape is read, the desired lines and characters are displayed on a specialized cathode ray tube. Sensitized paper and/or film is exposed to the display and the results are developed and returned to the programmer.

The obvious application of the S-C 4020 is the rapid production of labeled graphs. Results in graphical form are usually much easier to analyze than are the same results in printed tabular form.

Less obvious, but often highly desirable, is the exploitation of the S-C 4020 as a high-speed printer. For certain applications, this affords distinct advantages over the use of a standard printer. For one thing, printed records can be obtained on 35mm film, which greatly reduces the amount of space required to store the records. A form suited to the printed information can be generated or superimposed by the S-C 4020.

A variation is the production of a frame that combines tabular data with a plot of the related curve. Still another use of S-C 4020 techniques is the creation of diagrams and line drawings.

DESCRIPTION OF MACHINE FEATURES

Basic Device

The CHARACTRON shaped beam tube, the basic device of the S-C 4020, permits generation of certain characters at a very high speed. This tube contains a built-in character-forming section which consists of an electron gun, character-selection plates, and a stencil-like matrix.

The matrix, a small thin disk of special alloy metal, employes 64 character-shaped openings engraved in a space less than one-fourth of an inch square. These 64 characters are arranged in an 8 x 8 array.

The electron gun directs an electron beam toward the matrix through two sets of electrostatic character-selection plates. These plates deflect the beam so that it passes through the particular matrix aperture desired. Thus, as the beam leaves the matrix, it has been shaped in the form of the character aperture through which it has passed. As this shaped beam proceeds toward the tube face, it is accelerated (brightened) and again deflected to the proper position on the tube face. From this description, it can be seen that it takes no longer to print an entire CHARACTRON (beam-shaped) character than it does a single dot.

Each CHARACTRON character occupies approximately 6-horizontal by 9-vertical positions on the plottable area of the tube face. This area, called the raster, contains over one million plottable positions. These CHARACTRON characters cannot be rotated nor changed in size in any way. However, if other characters which may differ in size, shape, or attitude from those provided by the matrix are desired, they may be constructed from dots or lines.

Typewriter Simulator

A typewriter simulator, so-called because it provides automatic spacing and "carriage return" capabilities, is built into the S-C 4020. Each CHARACTRON character "typed" occupies 8-horizontal and 16-vertical positions on the raster; this character space area provides the spacing necessary to separate adjacent characters, both vertically and horizontally. In this mode of operation, a maximum of 128-horizontal and 64-vertical alphanumeric characters are possible in one display. The point at which the first character is to be printed must be specified (the center of the character position), and thereafter the horizontal printing position is automatically advanced one character-space to the right. At the edge of the frame, the horizontal position is returned to the left side of the frame and the vertical position is advanced down one character-space. This continues until the machine is told to stop printing, i.e., leave the typewriter mode of operation.

The typewriter-like operation takes place at an extremely rapid rate. Unlike the human typist, however, the machine does not check to make sure that it ends a line at the end of a word; a carriage return must be programmed or the appropriate number of blanks inserted to insure that the lines will come out correctly. Also, if the end of the film frame is reached, "typing" will continue at the upper left-hand corner of the current film frame (on top of anything that was put there before, producing a heavy black "burn") unless an advance-film command is given.

Variable Intensity

Heavy and light intensities of exposure are available for recording CHARACTRON-generated characters. A greater range of intensities may be obtained by repeated exposure at either light or heavy setting. This control does not affect the intensity of lines drawn by the axis generator or the vector generator. Over-exposure due to excessive repetition results in haloing on the film and poor definition in the output. Once the intensity has been set, all plotting and typewriter printing will be at that intensity until it is changed. For special applications, a variable intensity plot command option is available. This permits plotting at any of sixteen intensity levels without affecting the intensity mode set by a light or heavy command.

Vector Generator

A vector generator permits drawing a line from a specified point to the resultant of the specified X and Y components. The length of one vector is limited; the X and/or Y components may be no greater than 63 raster counts*. A longer line may be formed by a series of vectors plotted head-to-tail. Since the extent of a vector is determined by the X and Y components, the vectors will not necessarily stop at the edge of the frame. This feature might be used to advantage in marking the location of errors. The vector generator facility offers a method of forming characters which may be varied in size and in orientation. Subprograms are available for the generation of vector characters.

Axis Generator

An axis generator enables grid lines to be drawn horizontally or vertically across the film frame. The coordinates of the starting point of each axis must be specified. The stop point is specified as the Y coordinate or the complement of the X coordinate, whichever is appropriate.

Cameras

A 35mm microfilm camera (with sprocket holes) is available at GSFC. The image size is a 17.5-mm square on 35mm film. Also available at GSFC is a

*The total number of plottable positions in either the horizontal or vertical direction is 1024 "raster positions."

9.5-inch (hard copy) camera which records the same picture on a 7.5-inch square of sensitized vellum paper.

Rotatable Tube Mount

A rotatable tube mount allows output to be obtained in any of four orientations on the film.

Form Projector

A form projector permits an overlay of a master form on top of the image projected from the CHARACTRON tube. This form is obtained by using a glass slide, independent of the tube, to project an image in registration with the tube output. The program can control when the form is "flashed" (projected), but the S-C 4020 operator must manually change the slides.

Since preparation of the master is an expensive procedure (it is not a 35mm slide), such a form is suitable only for certain types of jobs. For example, in some cases, the form cannot be program generated, or it is repeated so often that the machine time saved by a master justifies its cost. Refer to Appendix E for the Form Slides available at GSFC.

Specified Image Size Generator

A specified image size generator allows the image to be expanded (and returned to normal four inches) under program control. The size of the expanded image may be adjusted by the Customer Engineer. However, machine alignment in expanded image is difficult. This feature is used to abut successive frames of 9.5 inch hard copy output.

Current Point Register

A current point register in the S-C 4020 is used to hold the coordinates specified by the last operation. This register can be set by the programmer (by plotting a blank, if necessary, at the point desired), but not retrieved by him.

Input

Operating off-line, an F-53-1A buffer is normally used to control the tape drive and to convert tape records to 36-bit control words for the S-C 4020. The F-53-1A also provides the means for accepting high-density, blocked information from the tape unit and transmitting the information to the S-C 4020 on demand.

ERROR MARKS

Certain types of errors are indicated on the frame by one of the following error marks.

Void Mark

This is a fuzzy 3/4" diameter black mark which the S-C 4020 places in the margin when forced to proceed past an error condition. This occurs whenever a parity error in reading the tape cannot be corrected by rereading, or whenever the tape block exceeds the size of the F-53-1A input buffer. In either condition, the S-C 4020 will automatically try a second time to read the record. The void light comes on only when the operator "forces" the S-C 4020 to proceed after detecting an error.

Error Symbol

The CHARACTRON character □ is displayed by the S-C 4020 if a parity error occurs when printing or plotting; that is, the specifications for a symbol are garbled. It is positioned at the coordinates contained in the erroneous word. This error symbol is not displayed if the erroneous word is one which normally does not display a character; for example, an axis or vector generation word. In addition to displaying the error symbol, the tape drive is stopped when a parity error is detected. A frame with such a mark will usually have a Void Mark as well.

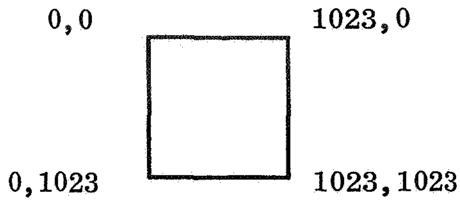
```
               86□3□85
               8780.□9
Error Symbol   88868.6
               8957.29
               9046.86
```

POSITIONING

Hardware Orientation

The S-C 4020 plots on a grid of 1024-horizontal by 1024-vertical points. There are, therefore, over 1 million addressable positions. The center of any character may be plotted at any of these positions. However, each character in the matrix occupies an amount of space that varies from the square of 3-horizontal by 3-vertical points used by a plotting dot to the rectangle of 8-horizontal by 16-vertical points used by all characters in the typewriter mode.

The S-C 4020 grid is a reflection of the first quadrant. The origin (0, 0) is in the upper left corner, with X and Y increasing to the right and downward, respectively as shown in the figure below. <u>This is the machine-oriented reference</u>. (See <u>Positioning Information</u> which follows regarding programmer orientation of rasters.)

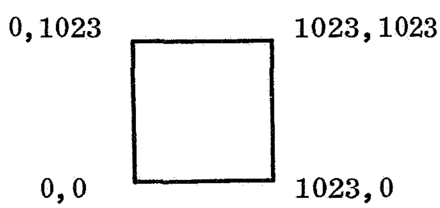


The plot appears in a 4-inch by 4-inch square on the face of the CHARACTRON shaped beam tube and is recorded on film.

Information is transmitted to the S-C 4020 through a 36-bit input register. The basic unit breaks down each 36-bit word into its component parts: the operation code and the information necessary to accomplish the specified operation. The bits, from left to right, are numbered from 0 (or S, the sign bit) through 35. Word format and operation code for each command is given in the following pages of this section.

Positioning Information

For all the routines discussed in the following sections, unless otherwise noted, the raster is reflected about the X axis to place the origin (0, 0) in the lower left-hand corner and (1023, 1023) is in the upper right-hand corner. The subroutine does the transformation to the S-C 4020 system described in the preceding Hardware Orientation. Any position in the frame can be addressed by specifying the number of raster counts from the left and from the bottom of the frame. Fractional distances cannot be addressed. The programmer raster orientation is shown in the figure below.



PERIPHERAL EQUIPMENT

The F-53-1A off-line tape unit buffer is designed to serve as a buffer between the S-C 4020 and its data source, a magnetic tape unit. This unit will accept data at any rate from an IBM 729 IV tape unit.

The F-53-1A is equipped with core storage of 4096 seven bit characters. The operation of the F-53-1A is block sequential. Under control of the F-53-1A, a block of data is transferred from tape to core. The tape unit is then stopped and restarted after the first block of data has been processed and transferred to the S-C 4020. Longitudinal and lateral parity are checked during the transfer from tape to core. A detected error will initiate an automatic re-read of the complete record block. A machine stop will occur if a read error repeats. Re-read and/ or proceed may be executed manually by the operator after a machine stop.

The configuration of GSFC's F-53-1A is as follows:

> A 4096 character core - 1401/1403 simulator. The 1401/1403 simulator makes it possible to read BCD print tapes prepared for the 1403 printer and record the information on microfilm with the S-C 4020. The F-53-1A will sense the single-space, double-space, page-eject control codes on the 1403 tape, but will not sense the intermediate skips used with the carriage control tape on the 1403. The block lengths on the tape must be less than 1020 characters.

NOTE: The S-C 4020 configuration at GSFC cannot plot using <u>extra</u> high density output tapes.

CONTROL OPERATIONS

No information is plotted on the grid as the result of any of the following nine operations:

Code	Name
46	ADVANCE FILM*
46,56	RESET*
52	CARRIAGE RETURN
41	SELECT CAMERA 1*
42	SELECT CAMERA 2*
43	SELECT BOTH CAMERAS*
44	EXPAND IMAGE*
45	REDUCE IMAGE*
12	STOP TYPE

Advance Film

The ADVANCE FILM command causes the film in the camera (or cameras) selected to be advanced one frame. If the supply of film is exhausted, the command cannot be executed. An alarm is activated.

ADVANCE FILM* OPCODE 46

OP CODE	NOT USED
0 – 5	6 – 35

*NOTE - The S-C 4020 requires a delay following receipt of certain operation codes. The delay is of concern only when operating without an F-53 or other input buffer. These operations, presented in the control operations discussion, are marked with an asterisk.

Reset

The RESET command performs simultaneously the functions of the ADVANCE FILM, STOP TYPE, and EXPOSE HEAVY commands. It also resets the deflection registers. In the typewriter mode, operation code 56 may replace any of the character codes shown below. Any bits following the code 56 are ignored.

RESET* (PLOTTING MODE) OPCODE 56

OP CODE	NOT USED
0 5	6 35

RESET* (TYPEWRITER MODE) OPCODE 56

CHAR CODE	CHAR CODE	CHAR CODE	CHAR CODE	CHAR CODE	CHAR CODE
0 5	6 11	12 17	18 23	24 29	30 35

Carriage Return

The CARRIAGE RETURN command is effective only in the typewriter mode. Operation code 52, replacing any of the 6-bit character codes shown below, causes an effective typing position at the left of the grid and down one line (X=0, Y=Y last +16) to be assumed. Spacing, therefore, may be achieved through the appearance of this code in contiguous 6-bit groups. If a CARRIAGE RETURN command is given which would specify an effective position of X=0, Y=1024, the position assumed is X=0, Y=0 of the same grid, since Y is computed modulo 2^{10}. Typing is resumed with the next legitimate character.

CARRIAGE RETURN OPCODE 52

CHAR CODE	CHAR CODE	CHAR CODE	CHAR CODE	CHAR CODE	CHAR CODE
0 5	6 11	12 17	18 23	24 29	30 35

Select Camera

The SELECT CAMERA commands are used to open the shutter of the camera or cameras designated and to close the shutter on the one not selected. If only one camera is mounted, it is not necessary to give a SELECT CAMERA command.

SELECT CAMERA 1* OPCODE 41

OP CODE	NOT USED
0 5	6 35

SELECT CAMERA 2* OPCODE 42

OP CODE	NOT USED
0 5	6 35

SELECT BOTH CAMERAS* OPCODE 43

OP CODE	NOT USED
0 5	6 35

Expand Image

The EXPAND IMAGE command increases the image size on the CHARACTRON tube face. The exact size is adjusted so that successive frames on the microfilm will abut to allow plots to extend continuously through several adjacent frames. The number of raster points remains the same and the CHARACTRON characters remain the same size; therefore, the space between characters is increased in expanded image mode. The image may be expanded uniformly in both directions in order to avoid distortion in graphs due to uneven scaling. The image remains expanded until the REDUCE IMAGE command is given.

EXPAND IMAGE* OPCODE 44

OP CODE	NOT USED
0 5	6 35

Reduce Image

The REDUCE IMAGE command causes the image to be reduced to four inches square and to remain at four inches until another EXPAND IMAGE command is given.

REDUCE IMAGE* OPCODE 45

OP CODE	NOT USED
0 5	6 35

Stop Type

The STOP TYPE command is used to return the S-C 4020 to the plotting mode from the typewriter mode of operation. Operation code 12 may replace any of the character codes shown below. Bits following the 12 are lost.

STOP TYPE OPCODE 12

CHAR CODE	CHAR CODE	CHAR CODE	CHAR CODE	CHAR CODE	CHAR CODE
0 5	6 11	12 17	18 23	24 29	30 35

RECORDING OPERATIONS

Information is placed on the grid by using one of the following nine commands. The length of time that characters, plotted or typed, are exposed results in their being recorded on film with heavy or light density. There is, however, no density-mode selection when drawing vectors or generating axes.

Code	Name
00	PLOT
04	EXPOSE LIGHT
02	EXPOSE HEAVY
20	TYPE SPECIFIED POINT
22	TYPE CURRENT POINT
6X, 7X	DRAW VECTOR
30	GENERATE X-AXIS*
32	GENERATE Y-AXIS*
50	PROJECT FORM*

Plotting Mode

The PLOT, EXPOSE LIGHT, and EXPOSE HEAVY commands are exactly the same in that they plot the character represented by bits 18 through 23 at the point specified by the X and Y deflections. However, code 00 (PLOT) operates in the density mode to which the S-C 4020 has already been set, while 02 and 04 set the machine to the EXPOSE HEAVY and EXPOSE LIGHT modes, respectively. All plotting or typing subsequent to the recognition of an 02 or 04 operation code is done in the density mode thus established. The EXPOSE HEAVY mode can be terminated only by the reception of an EXPOSE LIGHT code. The EXPOSE LIGHT mode can be changed to the EXPOSE HEAVY mode by either an EXPOSE HEAVY code or a RESET code.

PLOT — OPCODE 00

OP CODE	NOT USED	X DEFLECTION	CHAR CODE	NOT USED	Y DEFLECTION
0 — 5	6 — 7	8 — 17	18 — 23	24 — 25	26 — 35

EXPOSE LIGHT — OPCODE 04

OP CODE	NOT USED	X DEFLECTION	CHAR CODE	NOT USED	Y DEFLECTION
0 — 5	6 — 7	8 — 17	18 — 23	24 — 25	26 — 35

EXPOSE HEAVY — OPCODE 02

OP CODE	NOT USED	X DEFLECTION	CHAR CODE	NOT USED	Y DEFLECTION
0 — 5	6 — 7	8 — 17	18 — 23	24 — 25	26 — 35

Typewriter Mode

The typewriter mode is initiated by either a TYPE SPECIFIED POINT or a TYPE CURRENT POINT command. A STOP TYPE or RESET operation returns the S-C 4020 to the plotting mode.

In the typewriter mode, the grid of 1024-horizontal by 1024-vertical points may be regarded as containing 64 lines of 128 typing spaces each. Once typing has been commenced, it continues across the line until the 128th space has been filled or a CARRIAGE RETURN command has been executed. In either case, typing continues on the next lower line at the extreme left of the grid (X=0, Y=Y last + 16).

An ADVANCE FILM command must be given after the 64th line has been typed; otherwise, typing will continue on the first line of the same grid at X=Y=0 with resulting overlay. If vertical alignment of characters is desired, X_o should be either O or divisible by eight.

Note that, in the typewriter mode, character codes 12, 52, and 56 are not legitimate in that they do not cause a character to be typed but instead cause an operation to be performed (STOP TYPE, CARRIAGE RETURN, and RESET, respectively).

TYPE SPECIFIED POINT (FIRST WORD) OPCODE 20

OP CODE	NOT USED	X DEFLECTION	CHAR CODE	NOT USED	Y DEFLECTION
0 5	6 7	8 17	18 23	24 25	26 35

TYPE CURRENT POINT (FIRST WORD) OPCODE 22

CHAR CODE	CODE CODE	CHAR CODE	CHAR CODE	CHAR CODE	CHAR CODE
0 5	6 11	12 17	18 23	24 29	30 35

For both of the above commands, all words after the first have the following format:

(SUBSEQUENT WORDS)

CHAR CODE	CHAR CODE	CHAR CODE	CHAR CODE	CHAR CODE	CHAR CODE
0 5	6 11	12 17	18 23	24 29	30 35

One of the character codes of the subsequent word's format may be replaced by the operation code for CARRIAGE RETURN. Characters specified by succeeding bits are typed. The six bits following the last character to be typed must contain either the RESET or STOP TYPE operation code. Any bits still remaining in this word are lost.

The current point is defined as the point specified by the last operation performed, i.e., either the last point plotted or typed, or the origin of the last vector or axis drawn. It is advisable, therefore, with a TYPE CURRENT POINT command to make the first two characters typed blanks.

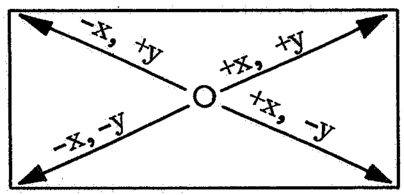
Vector Generation

The DRAW VECTOR command causes a vector to be drawn from the point specified by the X and Y deflections to the resultant of the X and Y component.

DRAW VECTOR OPCODE 1 BIT IN POSITIONS 0 AND 1

CODE	COMPONENT	X DEFLEC - TION	SIGN X COM- PONENT	SIGN Y COM- PONENT	COM - PONENT	Y DEFLEC- TION
0 5	6 7	8 17	18	19	20 25	26 35

The direction is determined by the signs of the X and Y components as shown below. A one in bit 18 indicates a plus X component; a zero means a minus X component. Bit 19 similarly indicates the sign of the Y component.



The maximum component of the vector which may be drawn is 1/16 of full scale (64 plotting positions).

Axis Generation

The GENERATE X-AXIS command causes a line to be drawn parallel to the X-axis. The line starts at the point designated by the X and Y deflections and extends to the right to the stop point. The ones complement of the X deflection of the stop point is placed in bits 6, 7, and 18 through 25 of the command word. If the stop point bits contain all zero, the axis generated will extend to the right boundary of the frame. Thus, programs prepared for early models of the S-C 4020 will generate full axes in machines equipped with the specified stop point axis generator. The stop point must be at least 64 raster counts to the right of the start point.

GENERATE X-AXIS* — OPCODE 30

OP CODE	STOP POINT	X DEFLECTION	STOP POINT	Y DEFLECTION
0 – 5	6 – 7	8 – 17	18 – 25	26 – 35

The GENERATE Y-AXIS command causes a line to be drawn parallel to the Y-axis. The line starts at the point designated by the X and Y deflections and extends upward to the stop point. The stop point is not complemented for a Y-axis. The Y deflection of the stop point is placed in bits 6, 7, and 18 through 25 of the command word. The stop point must be at least 64 raster counts upward from the start point. Programs pre-equipped with the specified stop point axis generator.

GENERATE Y-AXIS — OPCODE 32

OP CODE	STOP POINT	X DEFLECTION	STOP POINT	Y DEFLECTION
0 – 5	6 – 7	8 – 17	18 – 25	26 – 35

Form Projection

The PROJECT FORM command causes the information on the form slide to be superimposed on the file, along with the data displayed on the tube. This command may be given only once every 100 milliseconds.

PROJECT FORM* OPCODE 50

OP CODE	NOT USED
0 5	6 35

S-C 4020 OPERATION CODES

The 6-bit codes for commands, written in octal notation, are given below.

Operation	Code	Operation	Code
PLOT	00	SELECT CAMERA 2*	42
EXPOSE HEAVY	02	SELECT BOTH CAMERAS*	43
EXPOSE LIGHT	04	EXPAND IMAGE*	44
STOP TYPE	12	REDUCE IMAGE*	45
TYPE SPECIFIED POINT	20	ADVANCE FILM*	46
TYPE CURRENT POINT	22	PROJECT FORM*	50
GENERATE X-AXIS*	30	CARRIAGE RETURN	52
GENERATE Y-AXIS*	32	RESET*	56
SELECT CAMERA 1*	41	DRAW VECTOR	6x or 7x

OPTIONAL COMMANDS

Specified Intensity Plot

This command is an additional PLOT command. The intensity of the plotted character is controlled by regulating the exposure time. The sixteen levels of intensity are specified by setting bits 6, 7, 24, and 25 from 0000 (lightest) to 1111 (darkest). Bit 6 is the high order bit and bit 25 is the low order bit. The operation of this command does not affect the intensity mode for other plotting established by a prior EXPOSE HEAVY or EXPOSE LIGHT command. S-C 4020's not equipped with the Specified Plot Intensity option, treat this command as a PLOT with OPCODE 00.

SPECIFIED INTENSITY PLOT OPCODE 01

OP CODE	SPEC INT	DEFLECTION	CHAR CODE	SPEC INT	DEFLECTION
0 5	6 7	8 17	18 23	24 25	26 35

S-C 4020 Standard Character Matrix

The standard scientific character set used by the S-C 4020 is illustrated below.

S-C 4020 STANDARD SCIENTIFIC CHARACTER MATRIX

	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	8	9	∂	=	"	'	δ	α
2	+	A	B	C	D	E	F	G
3	H	I	π	.	)	β	±	?
4	–	J	K	L	M	N	O	P
5	Q	R	·	$	*	γ	~	d
6		/	S	T	U	V	W	X
7	Y	Z	°	,	(	∫	Σ	□

BCD Code in Octal = Number at Left and Above Character
Note: –12, 52, and 56 cannot be used in Typewriter Mode

SECTION II

SCORS II, SCORS IV AND LOCKHEED

SCORS II, SCORS IV AND LOCKHEED

As stated in the introduction of this document, the North American Aviation Package was the original utility package for the S-C 4020 Recorder at the GSFC. Due to its use and acceptance, it became known as SCORS II and the model for the subsequent SCORS IV and Lockheed packages. Because of their similarity, these systems are presented in one section.

When using the S-C 4020 in conjunction with the IBM 7094, particular attention should be given to what version of the FORTRAN compiler is being used for the source program's compilation. When the source program is FORTRAN II oriented, SCORS II must be used; but for FORTRAN IV compilations, SCORS IV is called upon. The Lockheed package, FORTRAN IV oriented, is available for use on the UNIVAC 1107/1108. For additional information, see Section IV, OPERATIONAL PROCEDURES, of this manual.

SCALING INFORMATION

The typical S-C 4020 application involves the display of some physical phenomenon scaled to the dimensions of the available plotting area. For the most part, the programmer can do his planning on the basis of the physical model. The system provides subprograms that will accept information given in terms of the model and perform the necessary scaling and conversion to relate the information to the raster area.

S-C 4020 CONTROL FUNCTIONS

INSERTION OF ID INFORMATION: IDFRMV

Prior to the selection of a camera at the onset of a job, the user should furnish the specific identification information for use within the ID frame via the subroutine IDFRMV. The calling sequence for the Lockheed package only is:

CALL IDFRMV (NAME, CODE, BLDG, PHONE, DATE)

The calling sequence for SCORS II and IV is:

CALL IDFRMV (NAME, CODE, BLDG, PHONE)

where all arguments are to be defined as Hollerith data (i.e., H format). The program requires two words (six characters each) for the name, and one word for each other argument. The date in the Lockheed package may range from 7H to 12H characters.

Lockheed package: If the IDFRMV subroutine is not used, the NAME and CODE fields of the ID frames are filled in from the contents of the first two fields of the run card and the DATE field is filled in with X's.

SCORS packages: Failure to use IDFRMV to insert information results in the insertion of dummy information within the ID frame.

SELECTING THE CAMERA: CAMRAV

At the outset of every job, both cameras are selected before drawing the ID frame. If no other camera selection is done all output will be on both cameras. A manual override is provided so that the S-C 4020 operator can change the camera selection at will.

Camera selection is implemented by the following statement:

CALL CAMRAV (N)

N=9 The 9-inch camera will be called and the 35mm camera will be dropped out of select.

N=35 The 35mm camera will be called and the 9-inch camera will be dropped out of select.

N= Any other number. Both cameras will be selected.

Note: With the Lockheed Package, as currently implemented, if N is coded as 16, the 35mm camera will be selected.

Note that selection of one camera causes the S-C 4020 display to be photographed only by the camera selected. That camera will be used until another CALL CAMRAV statement changes the selection. If the 9-inch camera is selected, and a CALL CAMRAV (35) is encountered, usage of the 9-inch camera will be discontinued and only the 35mm camera will be employed.

If both cameras are used alternately (rather than simultaneously) in a program, it is recommended that the following sequence be used as the first S-C 4020 statements in the program:

	CALL CAMRAV (N)	where N is any number other than 9,35, or 16, and
	CALL CAMRAV (9)	depending upon which camera is needed for the first display. That camera will be used until another camera is called for.
or	CALL CAMRAV (35)	

In this sequence, the first statement causes the simultaneous writing of identification frames on both cameras, since both are going to be used.

Therefore, CALL CAMRAV(935) is not equivalent to the two statements

CALL CAMRAV (9)

CALL CAMRAV (35)

ADVANCING THE FILM: FRAMEV, RESETV

Following the generation of the ID frame, a reset command is given by the program. This leaves the S-C 4020 with blank film in both cameras, set in the Expose Heavy mode, and not in the Typewriter mode. During the programming, it is the responsibility of the programmer to advance the film whenever a new page is to be started. A subroutine is provided which can be called by the statement:

CALL FRAMEV or

CALL FRAMEV(N)

N=0	The job number and the frame count for each camera will be displayed in the upper right corner of the frame.
N=Non-zero	Printing of job number and frame count will be suppressed.

Either call will advance the frame and update the frame count.

Another subroutine called by the statement CALL RESETV (N) also causes the film to be advanced. In addition, this command sets the exposure to heavy and assures that the S-C 4020 is not in the typewriter mode. The meaning of the parameter is the same as for FRAMEV. When drawing graphs and using the subroutine GRID1V to be described later, the film advance is controlled by a parameter of GRID 1V which in turn calls FRAMEV. No argument in FRAMEV or RESETV has the effect of a zero argument.

FRAME COUNT RETRIEVAL: NOFRV

NOFRV enables the programmer to obtain the current output frame counts. He can use this number as a "cut-off" point to prevent a wasteful loop. (Some programmers have used it, in conjunction with a timing routine, to determine how much computing time was needed to produce a frame, or for other accounting purposes within their programs.) The call statement is:

CALL NOFRV (N9, N35)

N9	Name of location in which current count of 9-inch camera frames will be stored.
N35	Name of location in which current count of 35mm camera frames will be stored.

Since both arguments represent output from the routine, they must be variables.

A special case of NOFRV is necessary to obtain the current frame count numbers printed (if different), instead of the current number of output frames on the S-C 4020. This occurs when the subprogram FRMNOV has been called. When a third argument is used in the calling sequence, the current camera frame counts appearing on the S-C 4020 output are returned in the first two arguments. The call statement is:

CALL NOFRV (N9, N35, ANY)

The format of the 3rd argument is immaterial.

INTENSITY SELECTION: BRITEV, FAINTV, RESETV

GRID1V ensures that the bright intensity mode is on. Normally, this intensity mode should be left on, since experience has shown that it produces the best results. If the programmer wants to change this setting to the faint mode, he can use the following statement:

CALL FAINTV

Then, to restore the bright intensity mode, he can use the statement:

CALL BRITEV

RESETV is explained under FRAMEV and STOPTV.

IMAGE SIZE CONTROL BIGV, SMALLV

The image size may be selected by the programmer to be 4.00 inches square or to be expanded, normally to 4.53 inches square. The program statement:

CALL BIGV

expands the image; and the statement,

CALL SMALLV

reduces the image. The system initialization program sets the image size to small.

PLOTTED CHARACTER INTENSITY: VINTV

Although this routine is a GSFC routine it is presented here as well as in Section B because of its use for intensity control.

This routine enables the programmer to control the intensity of plotted characters on the S-C 4020 Recorder according to a graded scale, from 1 to 16, designated in the calling sequence:

CALL VINTV (NC, IX, IY, INT)

where

NC	the character code (may be a decimal integer of position on grid or inform 1Hc).
IX, IY	raster count coordinates of point to be plotted, and
INT	the intensity designator, coded in the range of 1 to 16 (light to dark).

This routine exists in FAP for use on the IBM 7094 with the SCORS II package. It is also available on the 1107/1108 system libraries.

It should be noted that VINTV is a one-shot routine. It does not reset the standard intensity (the S-C 4020 does not permit this), and all other S-C 4020 routines will be plotted at the existing intensity setting (high or low).

PRODUCING MORE THAN ONE ID FRAME: RESIDV

RESIDV is a subroutine within the Lockheed package for the UNIVAC 1107/1108 which allows the programmer to produce more than one ID frame. The calling sequence is simply:

```
CALL RESIDV
```

where no arguments are required. This call generates another frame before another plot command is executed.

MODIFICATION OF INITIALIZATION OR TERMINATION PROCEDURE: CHAINV

A subroutine exists within the Lockheed package for modifying either the initialization or termination procedure of S-C 4020 plots. This routine is called CHAINV and has the following calling sequence:

```
CALL CHAINV (M,N)
```

where

M	Initialization argument, coded as M=Ø, Initialization procedure is not altered M=1, initial rewind is suppressed, ID frame is output M=2, both rewind and ID frame are suppressed.
N	Termination argument, coded as N=Ø, termination procedure is not altered N=1, trailing ID is output, one EOF written and no TRI N=2, empties plot buffer to tape but no ID, EOF or TRI N=3, same as 1 but trailing ID is suppressed.

When CHAINV (Ø, Ø) is called, an ID frame is generated at the next call to IDFRMV. Subsequent calls to IDFRMV will not output an ID frame again until CHAINV is called again. However, it will change the information on the trailing ID frame.

BYPASSING THE ID FRAME AND SPECIAL TREATMENT OF FRAME COUNTS: FRMNOV

FRMNOV serves two purposes. They are both for special applications and should not be of concern to the programmer for normal usage.

The first use of FRMNOV is to bypass the ID frame and initialization routine. This is done by calling FRMNOV within the plotting program, before issuing any other call that puts information on the S-C 4020 output tape. The frame normally following the ID frame will then have frame counts beginning with the arguments supplied in the CALL FRMNOV. SCORS IV counts the ID frame as frame 1 while SCORS II ignores the ID frame for counting purposes. This is normally 1 and 1 for each camera frame count. The call statement is:

CALL FRMNOV (M9, M35)

M9, M35	Normally these are both 1. Or they may be whatever frame counts are desired for the first (which is yet to be advanced) S-C 4020 output frame.

The second use of FRMNOV is to set either or both camera frame counts at any time in the program. If only one camera frame count is set, the other is unaffected. As in the above usage, the frame count setting made will appear on the following frame, not the frame currently exposed on the tube of the S-C 4020. The call statement is:

CALL FRMNOV (M9)	Set 9-inch camera frame count.
or	
CALL FRMNOV (M9, M35)	Set only 35mm camera frame count if M9 is negative. Set both camera frame counts if M9 is positive.

The frame count settings will apply to the next frame to be advanced.

NOTE: FRMNOV is not contained in the LOCKHEED package.

CHANGING THE SYSTEMS OUTPUT TAPE: TPNUMV

Within the SCORS II and Lockheed packages only, TPNUMV is used for special applications and should not be of concern to the programmer in normal usage.

The SCORS II system will write output on output tape A7 in the FORTRAN II version. The output tape for the Lockheed package is designated as H. This corresponds to Fortran logical unit 7 for both packages.

In the FORTRAN II version, the decrement of the location CRTAPE in deck OUTV must be set to the logical tape number of the desired output tape unit. In the IBJOB version, an output file must be redefined.

At object time, in FORTRAN II, the output tape can be changed with the FORTRAN statement:

```
CALL TPNUMV (LOGNUM)
```

which causes all successive plot outputs to be written on the tape designated by the logical number LOGNUM. The capability of changing the output tape during object time of an IBJOB run is not currently available.

EMPTYING THE OUTPUT BUFFER: PLTND, EOFTV

SCORS Packages: The S-C 4020 output buffer will not be dumped on tape until it is full unless the programmer forces it out for his own purposes. The current contents of the S-C 4020 output buffer can be dumped by the FORTRAN statement:

CALL PLTND

Since the buffers are not dumped until full, the last command of any program using the S-C 4020 programming system must be a CALL PLTND followed by a statement to write an END OF FILE on the output tape. The PLTND subroutine is not available in the Lockheed Package.

Lockheed Package: Although PLTND is not available within the Lockheed package, an end of job routine called EOFTV is contained within the package. This routine initiates a trailing ID frame, writes 3 EOF's and rewinds with interlock the output tape. The calling sequence is simply:

CALL EOFTV

(See also CHAINV Section II pg. 6)

B. GRAPHS

GRAPHING DATA: KWKPLT

The purpose of this routine which is available only in the SCORS packages, is to provide the programmer with a quick look at the relationship between two variables. KWKPLT will automatically provide the programmer with a series of linearly connected points of a scaled linear grid with or without identification printing.

The calling sequence without identification printing is:

CALL KWKPLT (LX, LY, N)

where LX — starting location of a forward stored array of floating point numbers representing the X-coordinates.

LY — starting location of a forward stored array of floating point numbers representing the Y-coordinates.

N — number of points to be plotted.

It is not necessary to arrange the coordinates in an increasing or decreasing order of magnitude. If the table of X-coordinates are not in ascending order, KWKPLT will rearrange them in ascending order within the table. The Y-coordinates will be arranged accordingly.

The calling sequence with identification printing is:

CALL KWKPLT (LX, LY, N, 18H(LH), 18H(LV))

where LX, LY, and N are the same as given above.

(LH) — 18 character identification for the X-coordinates.

(LV) — 18 character identification for the Y-coordinates.

The printing routine assumes a full 18 characters including blanks.

C. GRIDS

GENERATING A GRID: GRID1V

In many ways, plotting on the S-C 4020 is very much like plotting on a sheet of graph paper, but there are also distinct differences. For one thing, the programmer must create the grid; the film frame is completely blank to start with.

Although every line of the grid must be specified on the S-C 4020, there are advantages to this situation. A hand-plotted graph must be adapted to some pre-printed form; more frequently than not, this means that some plotting area must be sacrificed in order to use the most convenient scale.

On the S-C 4020, the programmer can select a scale that will be easy to read and that will accommodate the entire range of data. The number of light grid lines, the number of emphasized grid lines, and the spacing between lines can be chosen to suit the plot. The programmer is not restricted to the use of a single form for a variety of plots. For each graph, a new grid can be tailored to the data.

The easiest way to create a grid for S-C 4020 plots is to call the GRID1V subprogram. At the outset, GRID1V makes certain that the Typewriter Mode is off, and that the Bright Intensity Mode is on.

GRID1V will produce a grid which has some lines emphasized and some lines labeled. Margin space (which may be used for titles) will be reserved at the top, left side, and bottom of the grid. Normally, the title margin spaces are 24 raster counts wide.

Upon completion of GRID1V, scale factors will have been established and made available (internally) for the conversion requirements of other subprograms;

i. e., the conversion of floating point coordinates into raster coordinates.

The call statement for GRID1V appears below, with a description of the arguments.

CALL GRID1V (L, XL, XR, YB, YT, DX, DY, ±N, ±M, ±I, ±J, ±NX, ±NY)

L — This integer argument controls the film advance and frame identification display:

- L=0 — The film will be advanced, ID will be printed but the corner marks will be suppressed.
- L=1 — The film will be advanced. The job number and frame counts will be placed in the upper right corner of the frame.
- L=2 — The film will not be advanced and the identification information will not be displayed.
- L=3 — The film will be advanced but the identification information will be suppressed.
- L=4 — The film will be advanced and both the ID and the corner marks will be suppressed.

XL, XR — Floating point values of X for the left-most and right-most limits of the grid.

YB, YT — Floating point values for the bottom limit and the top limit of the grid.

After margin space for titles and labels has been reserved, the limits of the remaining space are assigned the data values given for XL, XR, YB, and YT. Scale factors are computed; they will remain in effect until another GRID1V statement is made (or until other action is taken to compute new scale factors).

DX, DY — Floating point data increment at which vertical (specified by DX) and horizontal (specified by DY)grid lines will be displayed. If 0. 0, no lines will be shown.

Positions are stepped off in DX increments in the positive and negative directions from X=0, and DY increments in the positive and negative directions from Y=0.

N, M — Fixed point integers that cause every Nth vertical grid line and every Mth horizontal grid line to be retraced for emphasis. If N (or M) is zero, no vertical (or horizontal) lines will be emphasized.

To force the grid to be square, a negative sign should be used on N and/or M. (If either N or M is zero, the negative sign should go on <u>both</u> N and M).

I, J — Fixed point integers which cause every Ith vertical line and every Jth horizontal line to be labeled. If I (or J) is zero, no vertical (or horizontal) lines will be labeled.

If I and J are positive, the line labels will lie along the X=0 and Y=0 lines, provided these lines are within the grid limits. If X=0 (or Y=0) does not fall within the grid limits, labels will be placed in a space reserved at the left (or at the bottom) of the grid.

Negative signs can be used on I and/or J to force labels outside the grid. Label space is reserved at the left if I is negative, or at the bottom if J is negative, and labels will be placed in these reserved spaces. Note that label margin space is in addition to the margin reserved for titles.

NX, NY — Fixed point integers indicating the number of characters to be displayed in the labels of vertical and horizontal lines.

+NX, +NY — The labels will be in a decimal format similar to the F-type format. In specifying +NX and +NY, a decimal point must be counted as one of the NX or NY characters, but the sign is not counted. The largest number of digits permitted is 6 (or 7 if one character is a decimal point).

-NX, -NY	The labels will be in scientific notation. (Example: $1.15\text{x}10^{+02}$.) NX indicates the number of significant figures in the labels of vertical grid lines, and NY indicates the same for the labels of horizontal lines. The sign, decimal point, and exponent will be displayed in addition to NX (or NY) characters. NX (or NY) must not be greater than 6.

Examples of GRID1V Usage

Figures 2-1 through 2-9 are examples of the effect of the various parameters in the GRID1V call statement. These examples are reproduced from S-C 4020 output. The call statement to produce each graph is printed by the S-C 4020 on the frame with the grid.

Figure 2-1 is a simple grid with the x=0, y=0 lines crossing in the middle of the grid. The numeric labels have been placed along the x=0, y=0 lines. For simplicity in the illustration, constants were used in the parameter list. In actual usage, variable names may be substituted for any parameter.

Figure 2-2 is similar to Figure 5-1 except that the XL, XR and YB, YT have been reversed to show that the scaling routines have no difficulty handling data which decreases from left to right and bottom to top.

Figure 2-3 illustrates the effect of negative values for I, J in the parameter list. Note that the numeric labels are outside the grid and that the margins have been increased to accommodate the labels.

Figures 2-4 and 2-5 show the same grid with labels in integer notation and scientific notation.

Figure 2-6 has been double exposed to show the effect of negative arguments at N or M. The outer frame was produced by the first call statement with the positive argument for N and M. The grid utilizes the maximum available space in both directions and is taller than it is wide. The second call statement with negative N and M forced the frame to be shorter in the Y direction in order to be square. It is important to have a square grid when representing geometric figures such as a circle or a square. Note that the first parameter of the second call statement is a 2 which inhibits the frame advance.

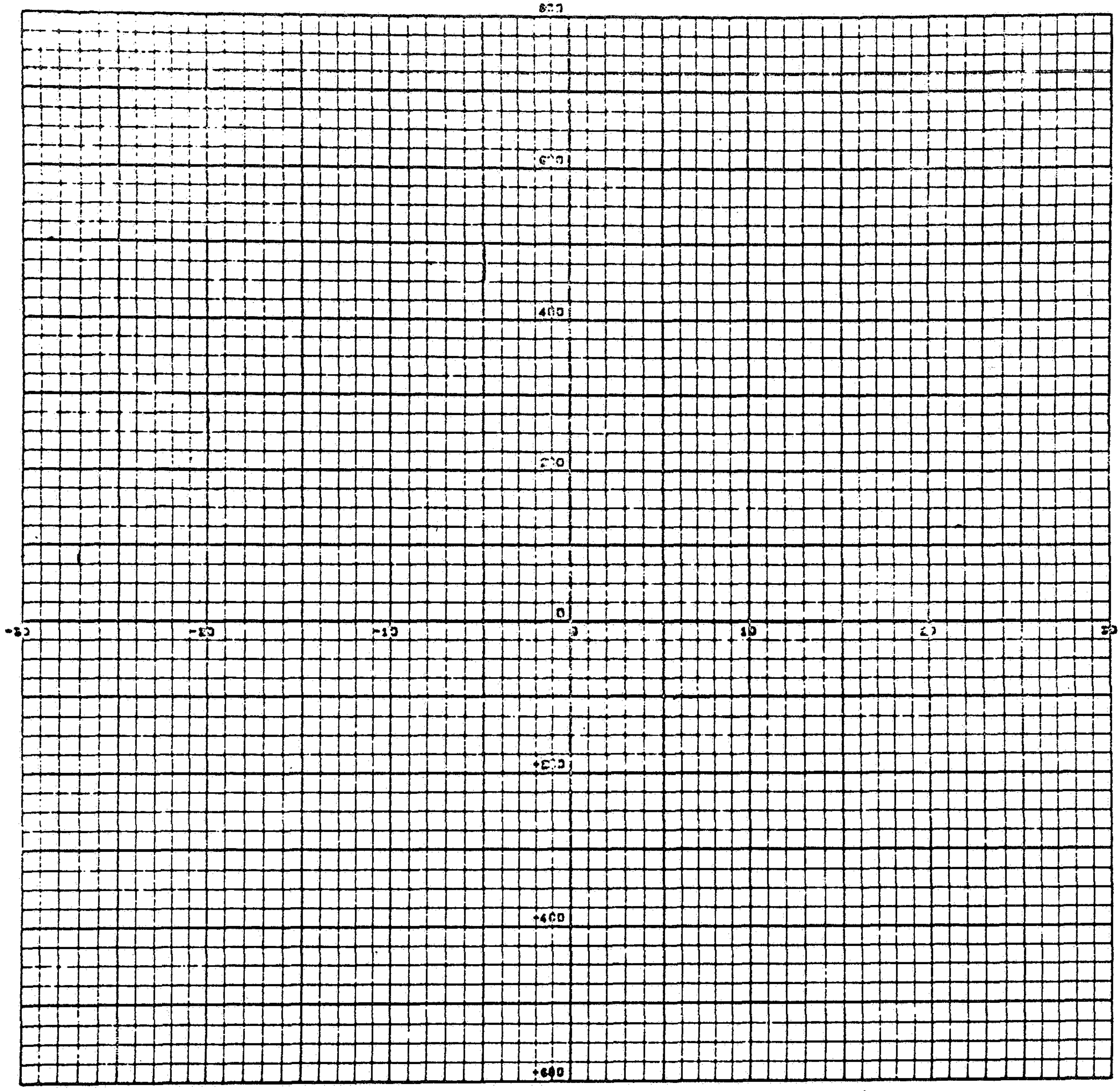
Figure 2-7 illustrates the use of the routine DXDYV to compute some of the values for the GRID1V parameter statement. DXDYV is explained below.

Figures 2-8 and 2-9 show the influence upon the grid of the density factor used by DXDYV. For the case of Figure 2-8, a density factor of 8.0 was specified as the 8th argument of DXDYV. A larger factor, 20.0, caused DXDYV to derive values of DX and DY such that the grid in Figure 2-9 is less dense.

COMPUTATION OF GRID1V ARGUMENTS: DXDYV

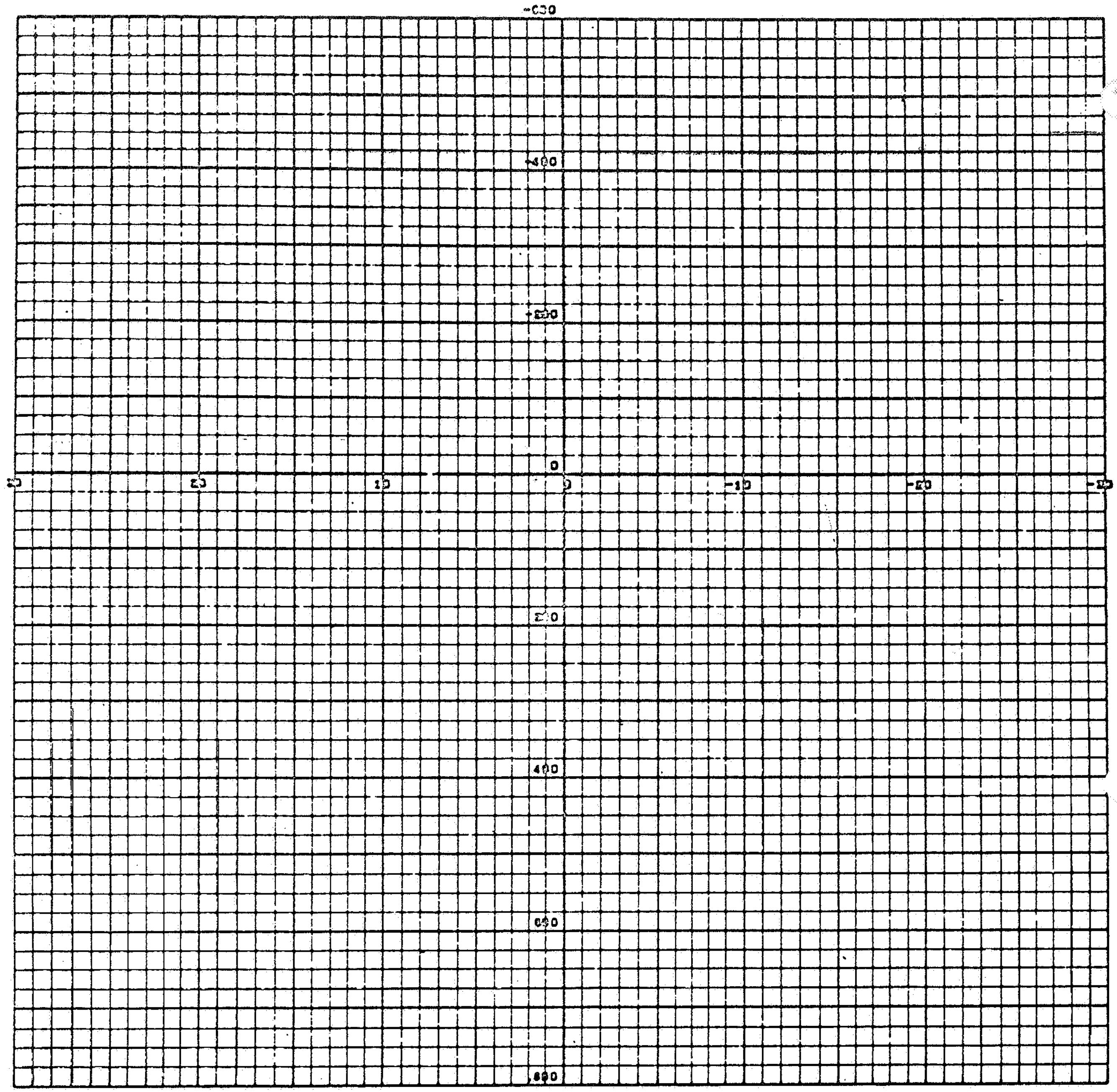
It frequently happens that the programmer does not have sufficient advance information about the range of data his program will encounter to be able to assign practical values to all arguments of GRID1V. In this case, a series of FORTRAN statements can be used to determine the upper and lower X and Y bounds. For example, the values of XL and XT for a block of data, X, can be computed as follows:

```
      XL= X(1)
      XR= X(1)
      DO 10 J= 2, NPTS       where NPTS is the number
                             of points in the X block of
      XL= MIN1F(XL, X(J))    data.
   10 XR= MAX1F(XR, X(J))
```



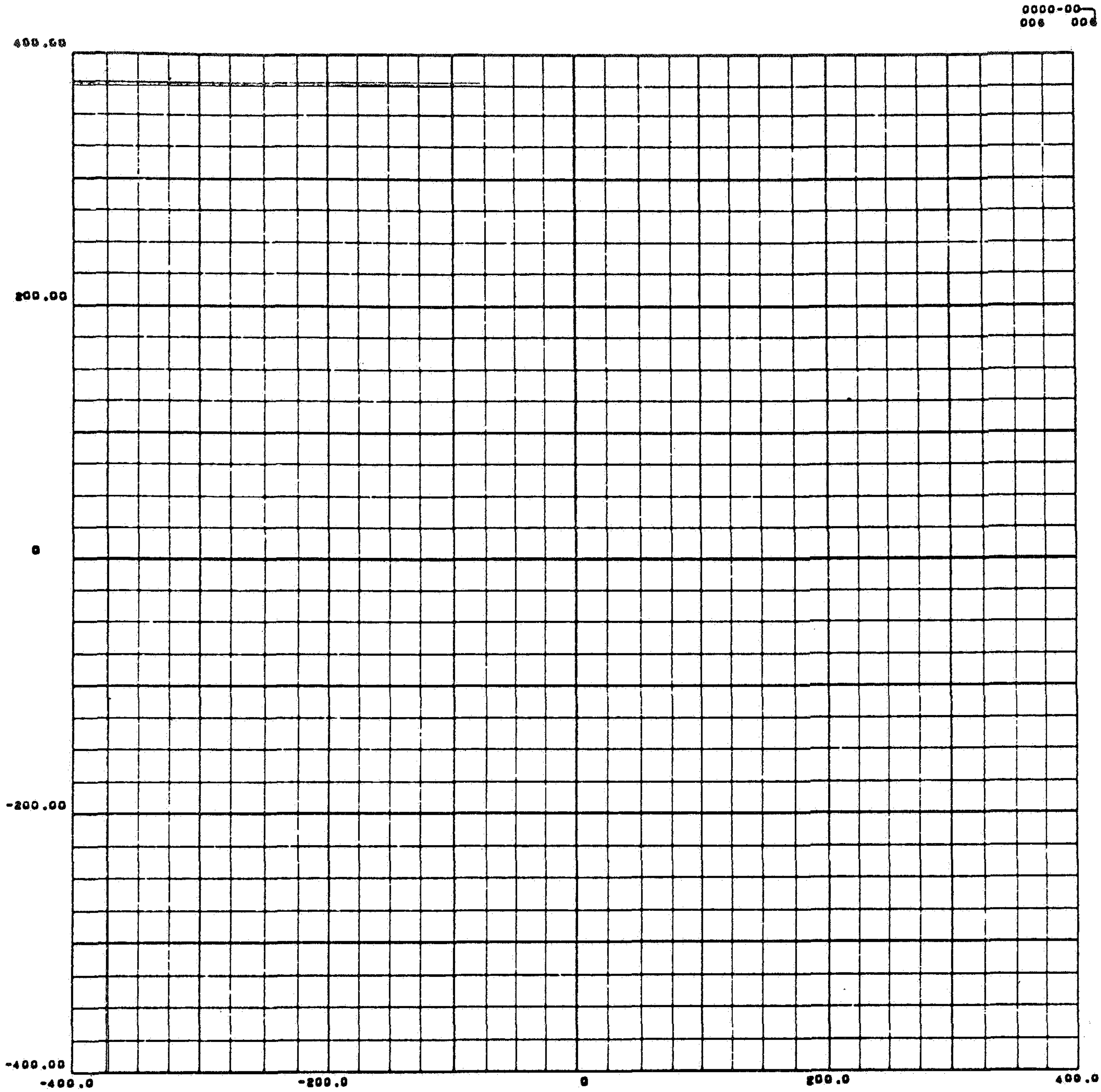
CALL GRID1V (1,-30.0, 30.0,-600.0, 600.0, 1.0, 25.0, 5, 4, 10, 8, 2, 3)

Figure 2-1



CALL GRIDIV (1, 30.0,-30.0, 800.0,-600.0, 1.0, 25.0, 5, 4, 10, 8, 2, 3)

Figure 2-2



CALL GRIDIV (1,-400.0, 400.0,-400.0, 400.0, 25.0, 25.0, 4, 4,-8,-8, 5, 6)

Figure 2-3

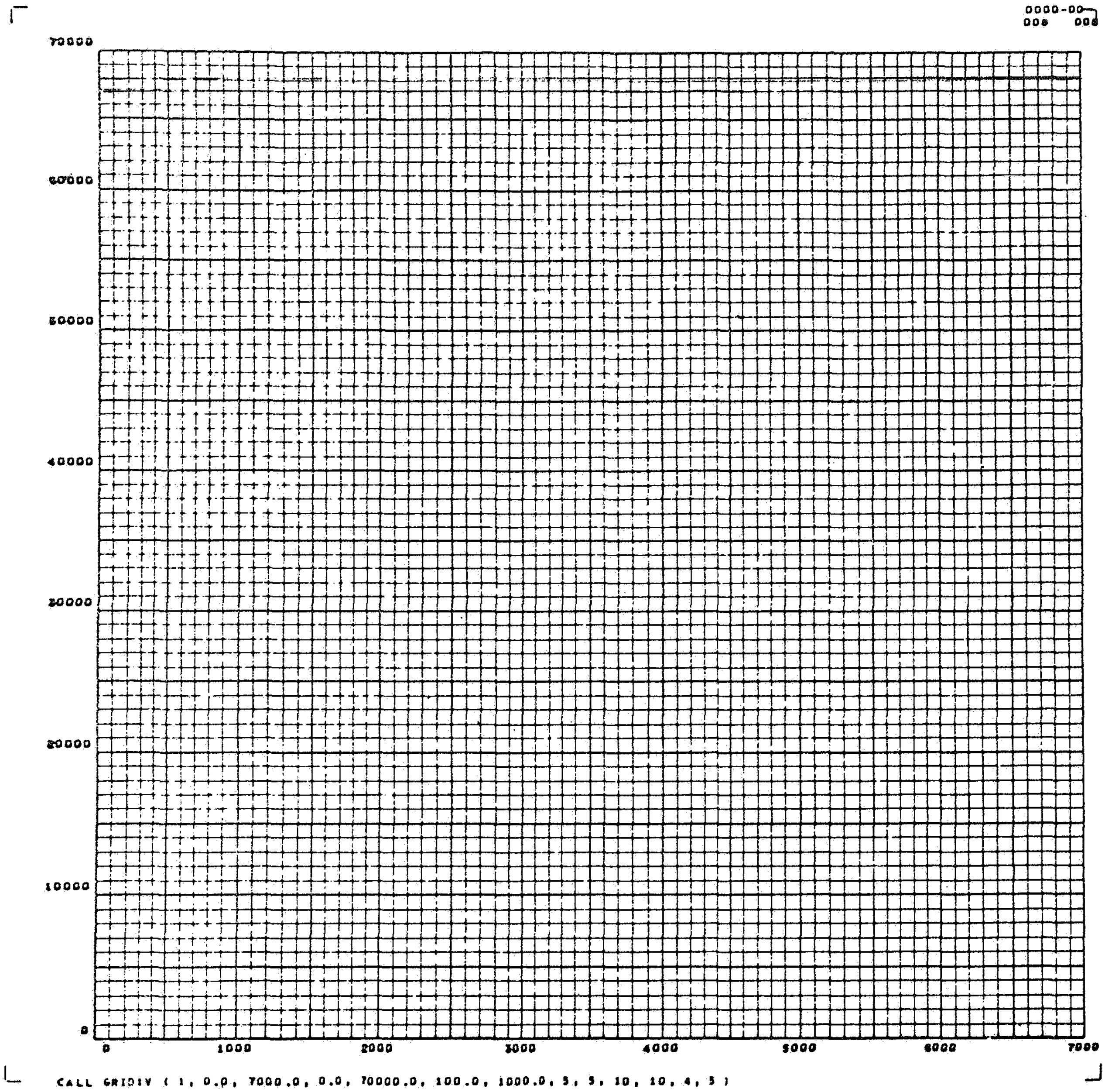

0000-00
008 008
70000
60000
50000
40000
30000
20000
10000
0
0
1000
2000
3000
4000
5000
6000
7000
CALL GRIDIV (1, 0.0, 7000.0, 0.0, 70000.0, 100.0, 1000.0, 5, 5, 10, 10, 4, 5)

Figure 2-4

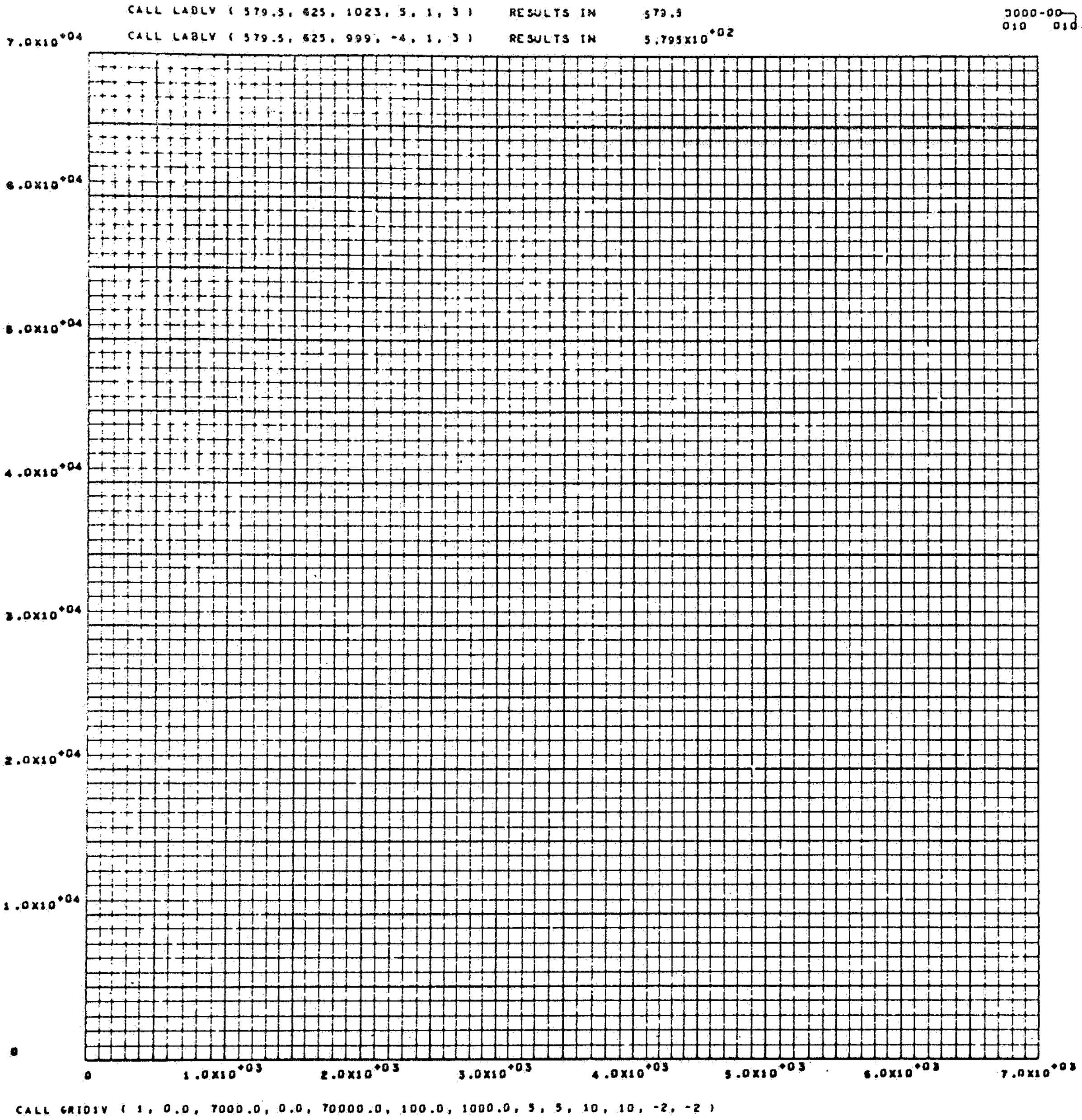
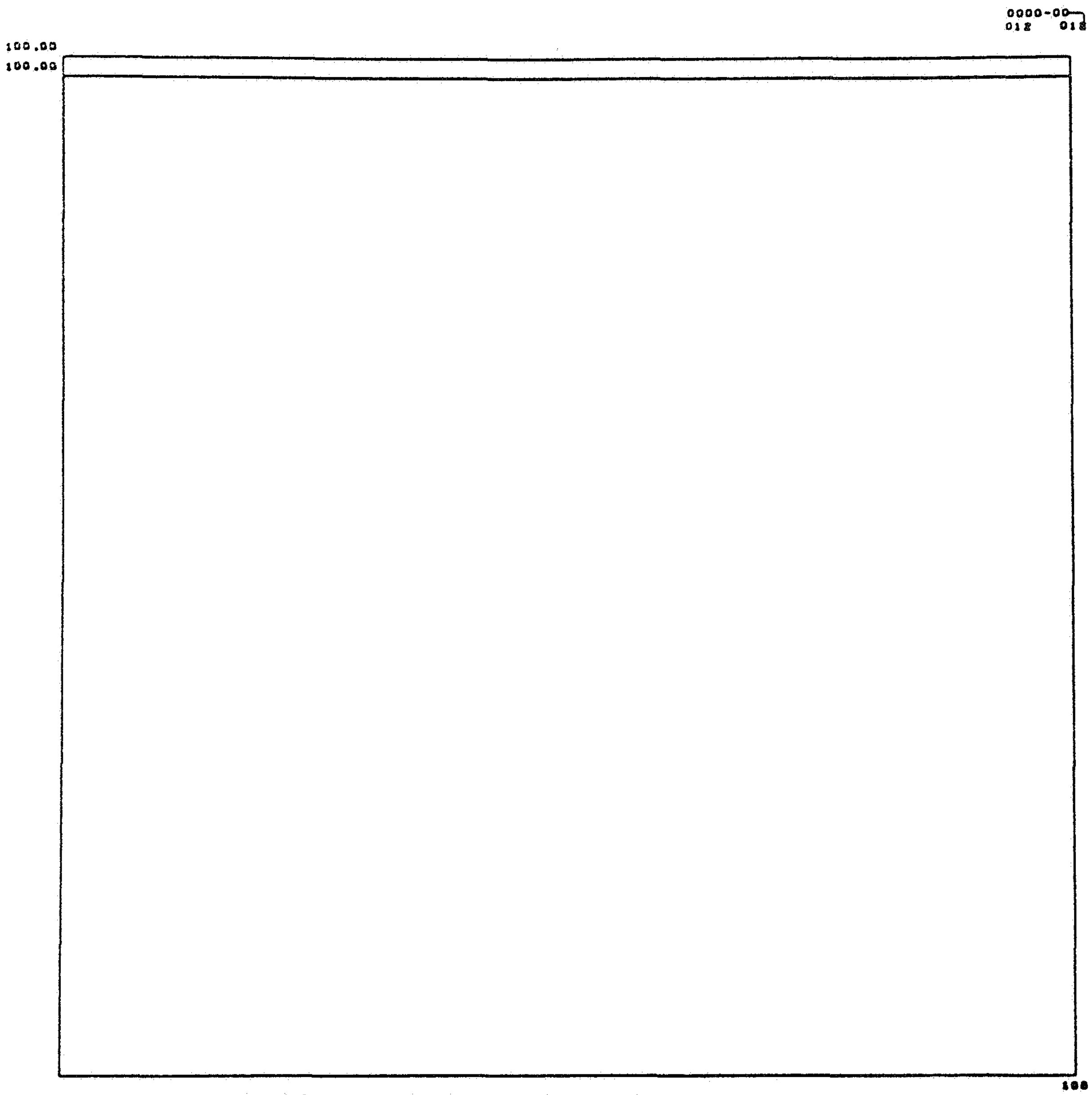
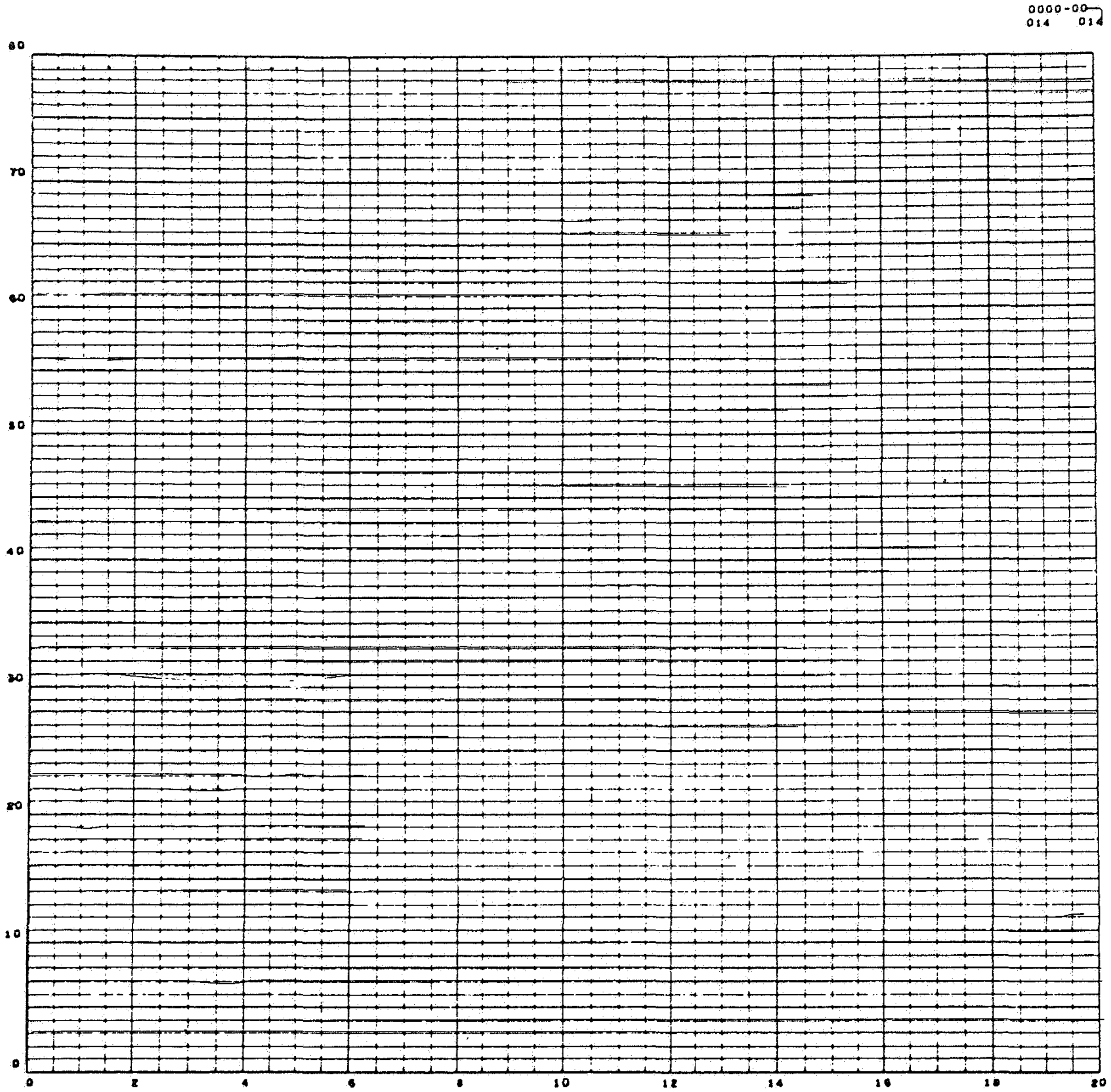

CALL LABLV (579.5, 625, 1023, 5, 1, 3) RESULTS IN 579.5
CALL LABLV (579.5, 625, 999, -4, 1, 3) RESULTS IN 5.795X10+02
7.0X10+04
6.0X10+04
5.0X10+04
4.0X10+04
3.0X10+04
2.0X10+04
1.0X10+04
0
0
1.0X10+03
2.0X10+03
3.0X10+03
4.0X10+03
5.0X10+03
6.0X10+03
7.0X10+03
CALL GRIDIV (1, 0.0, 7000.0, 0.0, 70000.0, 100.0, 1000.0, 5, 5, 10, 10, -2, -2)

Figure 2-5



CALL GRIDIV (1, 50.0, 100.0, 50.0, 100.0, 50.0, 50.0, 1, 1, 2, 2, 3, 6)
CALL GRIDIV (2, 50.0, 100.0, 50.0, 100.0, 50.0, 50.0,-1,-1, 2, 2, 3, 6)

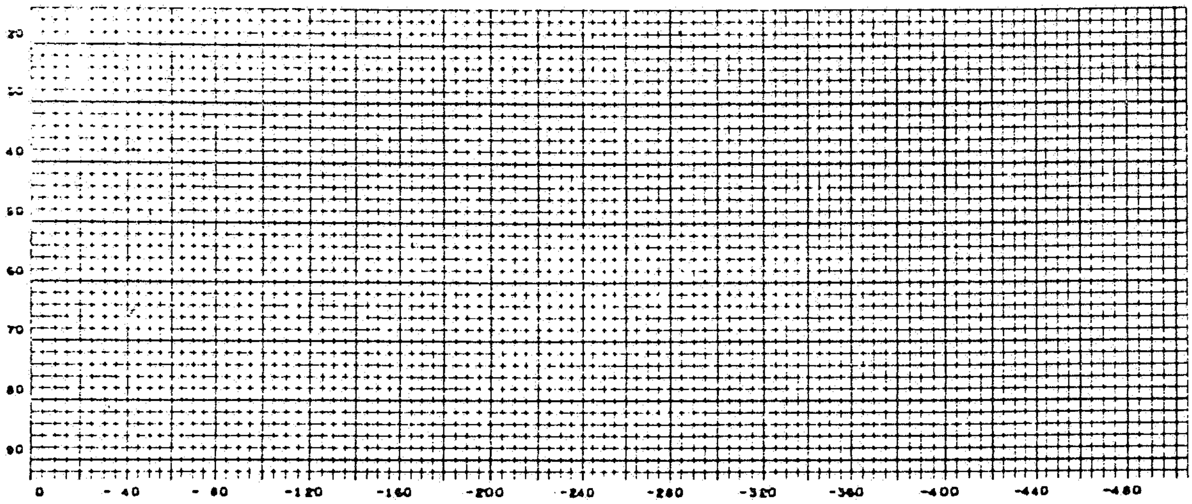
Figure 2-6




```
CALL DXDYV (1, 0.0, 20.0, DX, N, I, NX, 10.0, IERR1 )
CALL DXDYV (2, 0.0, 80.0, DY, N, J, NY, 10.0, IERR2 )
CALL GRIDIV (1, 0.0, 20.0, 0.0, 80.0, DX, DY, N, N, I, J, NX, NY )
```

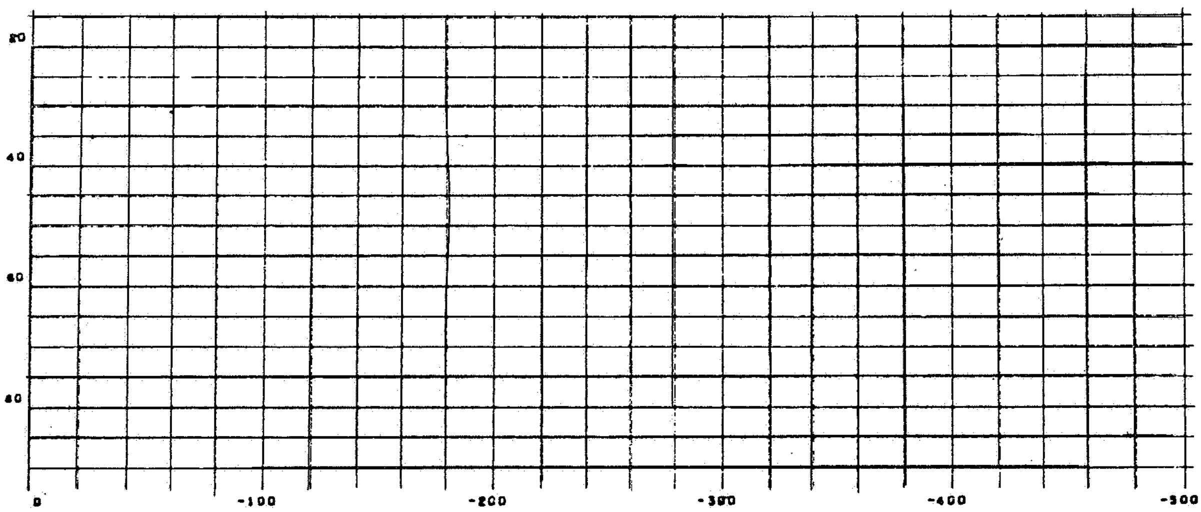
Figure 2-7

0000-00
016 016




```
CALL DXDYV ( 1, 0.0, -505.0, DX, N, I, NX, 8.0, IERR )
CALL DXDYV ( 2, 93.5, 14.0, DY, M, J, NY, 8.0, IERR )
CALL GRIDIV ( 1, 0.0, -505.0, 93.5, 14.0, DX, DY, N, M, I, J, NX, NY )
```

Figure 2-8




```
CALL DXDYV ( 1, 0.0, -505.0, DX, N, I, NX, 20.0, IERR )
CALL DXDYV ( 2, 93.5, 14.0, DY, M, J, NY, 20.0, IERR )
CALL GRIDIV ( 2, 0.0, -505.0, 93.5, 14.0, DX, DY, N, M, I, J, NX, NY )
```

Figure 2-9

A similar group of statements can be used to compute YB and YT for the Y block of data.

Once XL and XR (or YB and YT) are known, the routine DXDYV is available to compute arguments for line spacing, line emphasizing, and line labeling. Two call statements are available, one for the X direction and one for the Y direction. They are:

```
CALL DXDYV (1, XL, XR, DX, N, I, NX, DC, IERR)
CALL DXDYV (2, YB, YT, DY, M, J, NY, DC, IERR)
```

On each entry to DXDYV, four arguments are furnished by the programmer:

> The first argument is a 1 or a 2, to indicate whether DXDYV is being applied in the X direction or in the Y direction.
>
> XL and XR, N, I, NX and YB, YT, M, J and NY are defined as in the summary of GRID1V arguments.
>
> DC represents a floating point quantity which limits the density of the grid. The grid lines drawn by GRID1V, using arguments furnished by DXDYV, will be no closer than DC raster positions. DC should never have a value less than 3.0; values of 8.0 to 20.0 are recommended.

The remainder of the arguments are variables to which DXDYV will assign values. Any value previously assigned these variables will be destroyed during execution of the subroutine. <u>NEVER USE CONSTANTS FOR THESE ARGUMENTS.</u>

IERR is an error indicator. It is set to <u>zero</u> if a reasonable grid can be drawn, and to <u>one</u> if the parameters given would result in an impossible grid. After execution of DXDYV, IERR should always be tested before proceeding to draw the graph.

DXDYV has not been modified since the scientific label capability was added to the system. Until this modification has been made, the programmer who uses DXDYV and also wants scientific labels must include some extra provisions in his program.

For one thing, the values of NX and NY derived from DXDYV must be set to -NX and -NY by the programmer. Also NX and NY must not exceed 6. Because the scientific labels are longer (by at least seven characters), there is considerable danger that the labels of adjacent vertical grid lines will run into one another. Usually, this can be prevented by inserting the following statement between the DXDYV statement and the call to GRID1V.

```
I= 2*I
```

If the grid density indicator, DC, is large enough (say DC $\geq$ 15.0), it may be sufficient to space the labels without taking other action.

GRID1V CONTROLS

Certain features of the basic linear GRID1V can be altered by subprograms that control its internal operation. The subprograms can be classified as "set" and "retrieve" routines since they permit information to be set by the programmer and retrieved during execution of GRID1V.

The routines that furnish values different from those normally employed by GRID1V are:

SETMIV,	which allows the programmer to make nonstandard margin assignments. The companion routine called by GRID1V to retrieve margin values is SETMOV.
SETCIV,	which makes it possible to provide extra space for grid line labels. The companion routine is SETCOV.

Routines that furnish indicators recognized by GRID1V as signals to execute alternate branches are:

HOLDIV,	which assists in holding margins from graph to graph. HOLDOV is called to retrieve the indicators.
SMXYV,	which enables the programmer to select a non-linear mode of operation. The companion routine is MSXYV. These two routines are described under "Log and Semilog Plotting."

GRID MARGIN VARIATION: SETMIV, SETMOV

As discussed in an earlier section, GRID1V normally reserves a strip, 24 raster counts in width, at the top, left, and bottom of the grid, for the display of titles. For the many applications which require special margin widths, the subprogram SETMIV can be called to change the basic specifications.

One obvious application of SETMIV is to provide margin space for multiple lines of printed titles and headings. In addition, and perhaps even more important, SETMIV makes it possible to display more than one graph on a frame, or to display a graph with its accompanying text.

The standard GRID1V margin specifications can be altered by the statement:

```
CALL SETMIV (MTL, MTR, MTB, MTT)
```

Each argument is an integer which specifies, in raster counts, the width of one area to be reserved for a margin.

MTL	Width of area for left margin.
MTR	Width of area for right margin.
MTB	Width of area for bottom margin.
MTT	Width of area for top margin.

GRID1V does not necessarily use these <u>exact</u> values for the upper and lower limits of X and Y. It guarantees that the reserved space will not be overlapped, assigning additional space if required for label margins. After the total margin space has been reserved, the remaining area will be used for the grid.

If SETMIV is never called, GRID1V will use the values 24, 0, 24, 24 as MTL, MTR, MTB, and MTT, respectively. To return to a standard grid after the margins have been altered, restore the standard margin values by:

```
CALL SETMIV (24, 0, 24, 24)
```

The current values of MTL, MTR, MTB, and MTT can be <u>retrieved</u> by using the statement:

CALL SETMOV (MTLL, MTRL, MTBL, MTTL)

where the arguments are variables (never constants) to which SETMOV is to assign the current margin values. SETMOV was designed for use by GRID1V to retrieve current margin values; the programmer will rarely have reason to call it.

Examples: Figure 2-10 shows three grids with the SETMIV and GRID1V call statements used to produce them. The grid at the bottom was the first one displayed; a 1 was used as the first argument of the first GRID1V statement executed, in order to change the film frame. The other two GRID1V statements include a 2 as the first argument, to inhibit the film advance.

Note particularly the variation in the raster locations assigned by GRID1V to XL in each of the grids. This effect is caused primarily by the differences in the specification of NY (the last argument), which gives the number of characters to be displayed in the labels of horizontal lines. In each case, NY has been assigned a value just large enough to satisfy the needs of the grid. For the bottom grid, NY=1; for the middle grid, NY=3; and for the top grid, NY=5. Since margin space was reserved for labels of different lengths, the positions of the left limits, and of the corresponding values of X, vary noticeably. Such a nonalignment is often of no importance, but if it does matter, the programmer may have to make special provisions to force alignment.

PROVIDING FOR SPECIAL LABEL CHARACTERS: SETCIV, SETCOV

GRID1V computes the starting location of each label, taking into consideration the size of the characters used. If the labels are to be placed outside the grid, GRID1V assigns space for them, again taking the character size into consideration. Normally, labeling is done in CHARACTRON characters (via LABLV). If the programmer substitutes a non-system labeling routine for LABLV, it may be necessary to furnish adjusted character dimensions to GRID1V.

To state the dimensions of nonstandard label characters, use:

CALL SETCIV (IW, IH)

IW An integer which specifies, in raster counts, the allowance needed for the width of each label character.

IH An integer which specifies, in raster counts, the allowance for the height of a label character.

If SETCIV is never called, the indicator table contains IW=8, and IH=10. Obviously, if it is called, the arguments must be compatible with the size of the characters employed by the LABLV subprogram used.

GRID1V retrieves the values of the indicators by using:

CALL SETCOV (IWL, IHL)

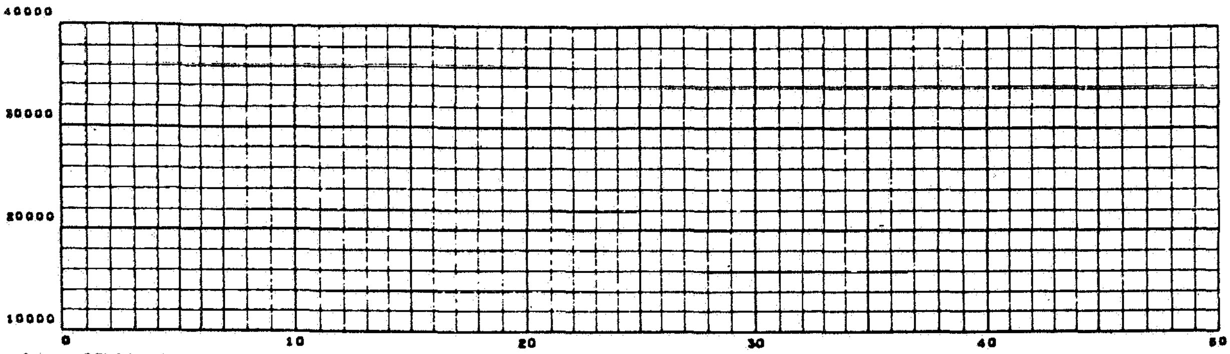
The width will be retrieved from the table and stored in the fixed point variable location IWL, and the height will be similarly stored in IHL. (The arguments must not be constants.) Note that GRID1V uses this information to control the space that will be reserved for labels; it does not control the size of the label characters themselves in any way.

HOLDING MARGINS FROM GRAPH TO GRAPH: HOLDIV, HOLDOV

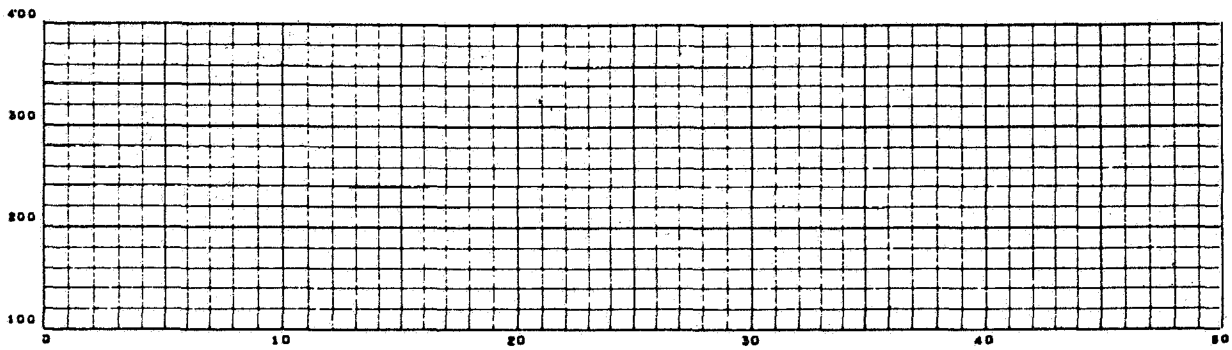
For a large graph, it may be necessary for the programmer to display segments of the graph in separate frames, and join the segments "tile fashion" to form the complete plot. If the graphs are to have the same scale, certain equalities should exist.

1. The range of X in each segment should be equal, and the range of Y should be equal. In other words, the quantity (XR-XL) should be the same in each segment, and, similarly, the quantity (YT-YB).

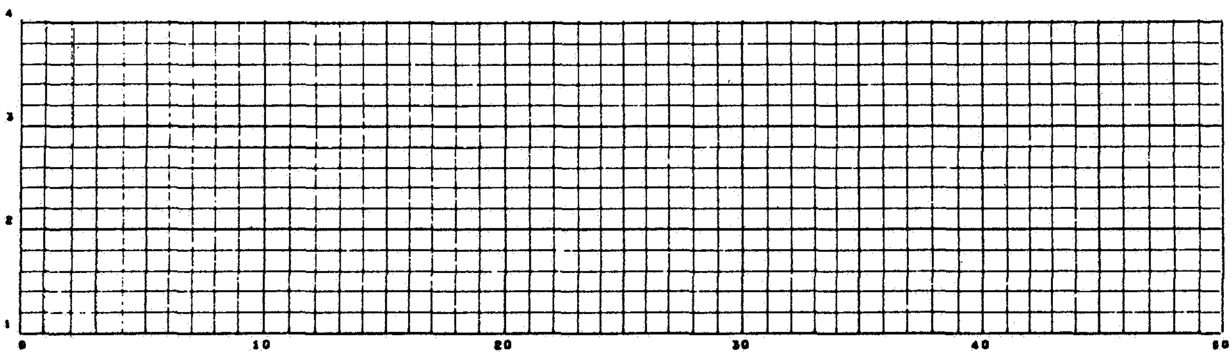
MARGIN VARIATIONS FOR GRIDIV




```
CALL SETMIV ( 24, 0, 712, 28)
CALL GRIDIV (2, 0.0, 50.0, 10000.0, 40000.0, 1.0, 2000., 5, 5, 10, 5, 2, 5)
```




```
CALL SETMIV ( 24, 0, 368, 371)
CALL GRIDIV ( 2, 0.0, 50.0, 100.0, 400.0, 1.0, 20.0, 5, 5, 10, 5, 2, 3)
```




```
CALL SETMIV ( 24, 0, 24, 715)
CALL GRIDIV ( 1, 0.0, 50.0, 1.0, 4.0, 1.0, 0.2, 5, 5, 10, 5, 2, 1)
```

Figure 2-10

2. The dimensions of the scaled area should be the same from segment to segment; that is, the dimensions of the space between margins should be equal. The programmer can easily provide for equality in the ranges of X and Y, but he cannot so readily ensure equality in the scaled areas. Since GRID1V computes label margins (and, therefore, total margins) to suit the needs of each graph, the dimensions of the scaled area may vary.

One method that will usually give equality of scaled areas is to specify the option that forces labels to be placed outside the grid, and to always request the same number of label characters (NX, YT) for each segment. If this is not practical, a "holding" feature is provided.

GRID1V can be instructed to hold the label margin spaces used for the preceding grid and use them in computing total margins for the next grid. The statement to be used is:

```
CALL HOLDIV (NH)
```

> If NH≠ 0, the label margins from the preceding grid will be used again. If NH=0 (as is the case if HOLDIV is never called), label margins will be computed in the normal manner.

The status of this indicator is tested in GRID1V by using:

```
CALL HOLDOV (NHL)
```

> The value of the indicator will be retrieved and stored in the location named in the argument.

The "hold" may be released by executing the statement:

```
CALL HOLDIV (0)
```

Figure 2-11 is similar to Figure 2-10 except that NY=5 on all three grids, permitting the left limits to be in line. If XL, NX, and NY are equal from grid to grid, the desired alignment will usually be achieved.

Figure 2-12 shows four graphs on a single frame. The SETMIV statements used to produce the margins for each grid are shown. The programmer must remember to set the first argument of the GRID1V statement to 2 so as not to advance the film.

Figures 2-13, 2-14, and 2-15 show additional examples of special effects which can be obtained with GRID1V when the routine HOLDIV is used to retain grid margins from one grid to another. The labeling is self explanatory.

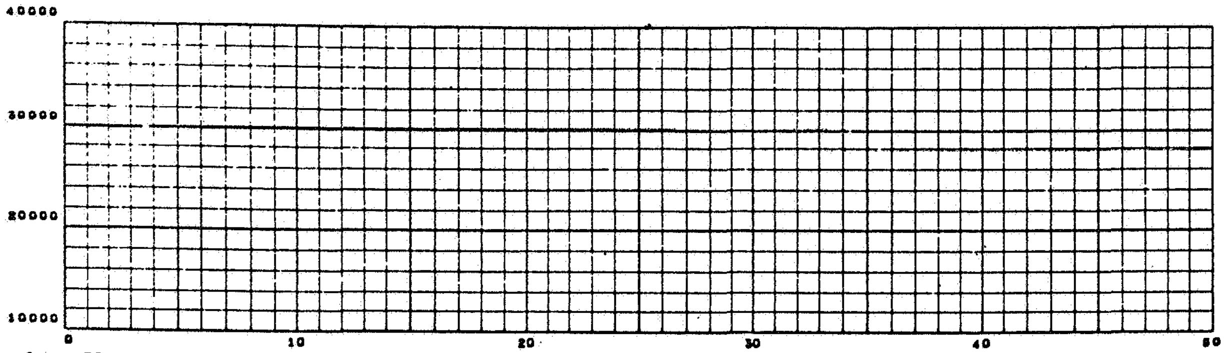
OPERATIONAL DETAILS OF GRID1V

GRID1V is, in many respects, an executive routine. It examines the information furnished by the argument list and by certain external subprograms, makes decisions based on the information, and then calls other subprograms to advance the film, compute scale factors, generate the grid, etc.

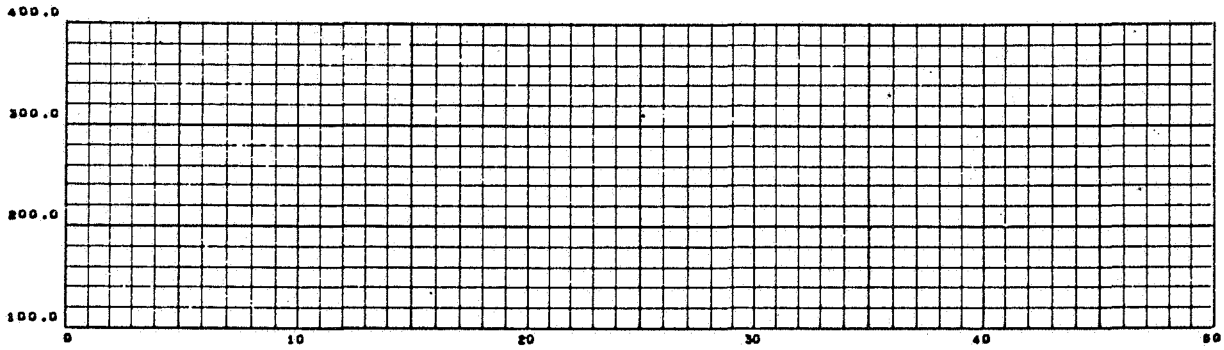
Initially, GRID1V uses STOPTV to ensure that the typewriter mode is "off", BRITEV to ensure that the bright intensity mode is "on", and FRAMEV to advance the film and affix the job number and frame count (unless an option is used that inhibits this).

GRID1V then checks certain internal locations to obtain basic information, by calling the following subprograms:

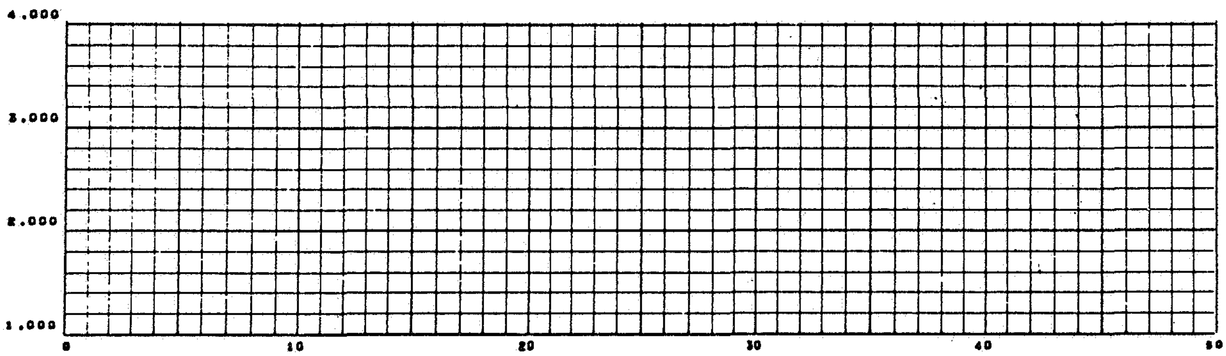
SETMOV	Retrieves margin assignments. The programmer may have changed the standard specifications by a CALL SETMIV.
SETCOV	Retrieves character size specifications that programmer may have changed by a CALL SETCIV.



CALL SETHIV (24, 0, 712, 28)
CALL GRIDIV (2, 0.0, 50.0, 10000.0, 40000.0, 1.0, 2000., 5, 5, 10, 5, 2, 5)



CALL SETHIV (24, 0, 366, 371)
CALL GRIDIV (2, 0.0, 50.0, 100.0, 400.0, 1.0, 20.0, 5, 5, 10, 5, 2, 5)



CALL SETHIV (24, 0, 24, 715)
CALL GRIDIV (1, 0.0, 50.0, 1.0, 4.0, 1.0, 0.2, 5, 5, 10, 5, 2, 5)

Figure 2-11

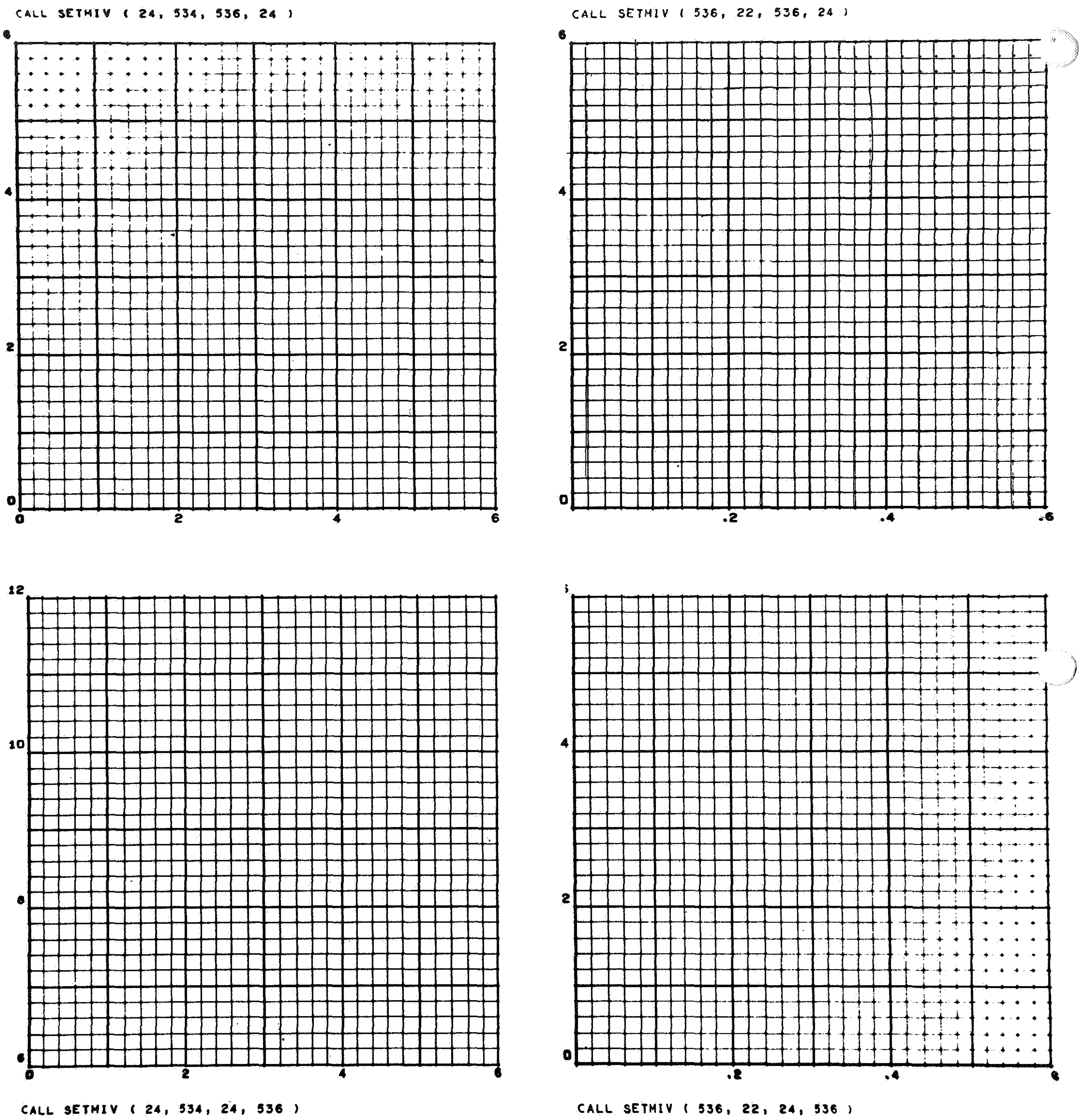
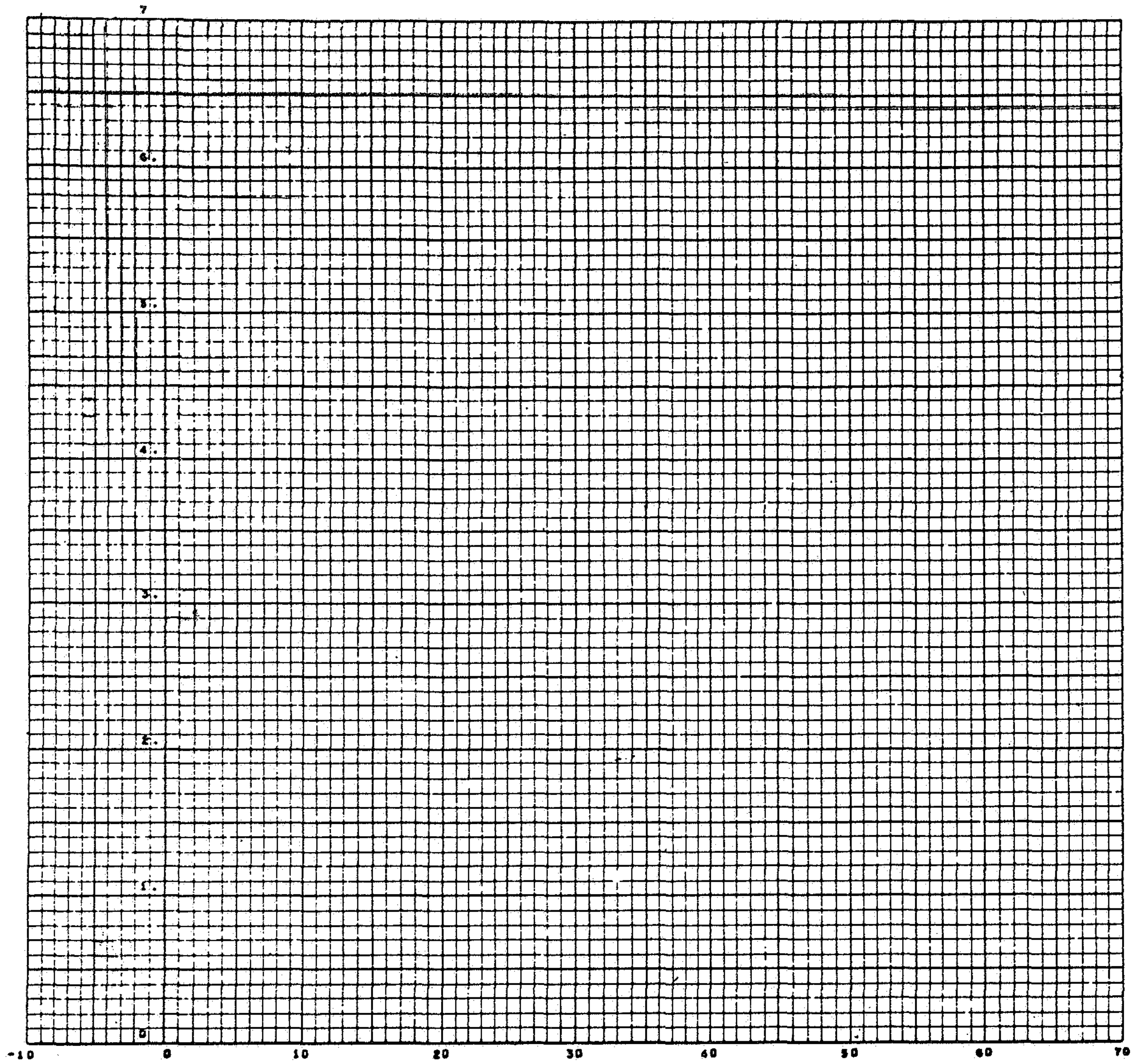

CALL SETMIV (24, 534, 536, 24)
CALL SETMIV (536, 22, 536, 24)
CALL SETMIV (24, 534, 24, 536)
CALL SETMIV (536, 22, 24, 536)

Figure 2-12

SPECIAL GRIDIV EXAMPLES

HOLDIV AS AN AID IN MAINTAINING THE SAME SCALE FROM ONE GRID TO ANOTHER

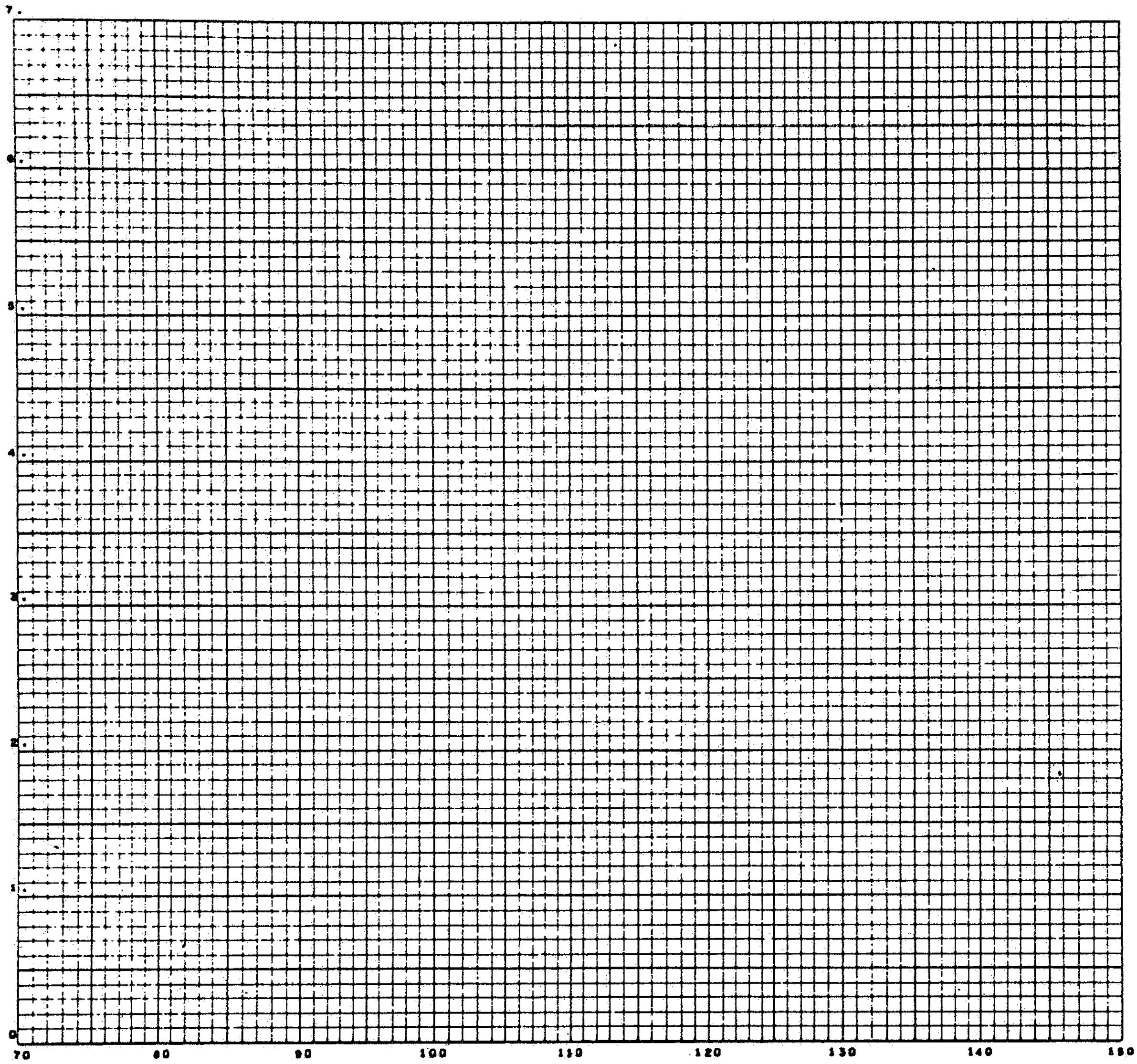


CALL GRIDIV (1, -10.0, 70.0, 0.0, 7.0, 1.0, 0.1, 5, 5, 10, 10, 2, 2)

CALL HOLDIV(1)

THE HOLD STATEMENT WILL CAUSE LABEL MARGINS TO BE HELD FOR SUBSEQUENT GRIDS
UNTIL CALL HOLDIV(0) RELEASES THE HOLD

Figure 2-13



CALL GRIDIV (1, 70.0, 150.0, 0.0, 7.0, 1.0, 0.1, 5, 5, 10, 10, 3, 2)
NOTE...NOT ONLY HAS HOLDIV BEEN USED BUT ALSO XL,XR,YB,YT HAVE BEEN SPECIFIED SUCH THAT
XR-XL AND YT-YB OF ONE GRID EQUAL THEIR COUNTERPART IN THE OTHER GRID

Figure 2-14

SPECIAL GRID1V EXAMPLES

USE OF HOLDIV WITHIN A FRAME

```
CALL SETMIV ( 24,599, 50, 573 )
CALL GRID1V ( 1, -10.0, 90.0, 0.0, 9.0, 5.0, 1.0, 4, 2, -4, -2, 2, 1 )
CALL HOLDIV (1)
CALL SETMIV ( 524, 99, 50, 573 )
CALL GRID1V ( 2, 90.0, 190.0, 0.0, 9.0, 5.0, 1.0, 4, 2, -4, -2, 3,1 )
```

NOTE... NOT ONLY MUST XR-XL AND YB-YT OF ONE GRID EQUAL THE CORRESPONDING QUANTITY IN THE OTHER GRID, BUT ALSO THE ARGUMENTS OF SETMIV MUST HAVE A SPECIFIC RELATIONSHIP........SAY THE ARGUMENT LIST IS (MTL,MTR,MTB,MTT) ,THEN THE VALUES 1023-(MTL+MTR) AND 1023-(MTB+MTT) OF ONE GRID MUST EQUAL THE CORRESPONDING QUANTITY IN THE OTHER GRID

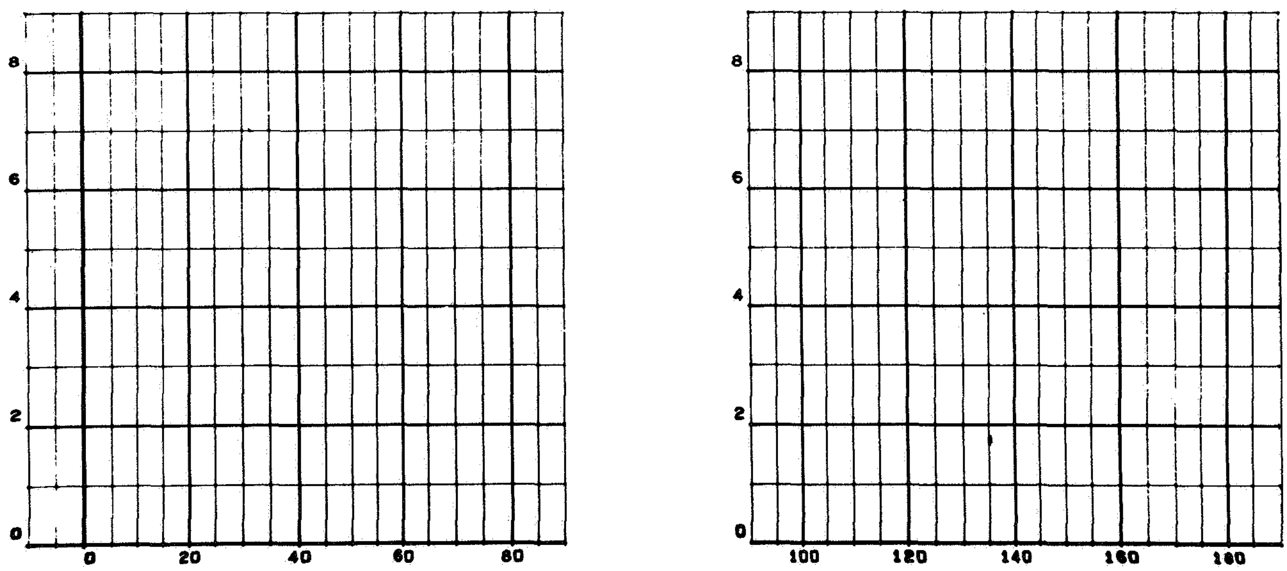


Figure 2-15

MSXYV	Determines linear or nonlinear mode. Indicators may have been set to nonlinear by a CALL SMXYV.
HOLDOV	Determines whether margin specifications were held over from preceding graph. Indicator may have been specified by a CALL HOLDIV.

Computations are made to determine the raster positions to be used as grid boundaries. ERRLNV and/or ERRNLV are called to check the boundaries and the data limits to see if a grid can be produced from this information. Finally, XSCALV and YSCALV are called to compute scale factors, and LINRV (in the linear mode) and/or NONLNV (in the logarithmic mode) are called to generate the grid.

DETERMINING GRID BOUNDARY POSITIONS

GRID1V sets aside the margin space specified by a prior call to SETMIV (or the standard margin space if SETMIV has not been called). With the exception of the small area in the upper right corner used for frame identification, GRID1V does no writing in these basic margins.

Then GRID1V tests to see if labels are to be placed (or may extend) outside the grid area. Space required for such labels is computed, taking into consideration the type of label (fixed point or scientific), the number of label characters specified in GRID1V arguments plus space for a sign, and the height and width of the label characters.

If any space is needed for labels at the left, right, bottom, or top of the grid, it is added to the basic margins specified by SETMIV table, to produce the total margins. (HOLDIV can alter this procedure by causing label spaces "held" from a previous grid to be added when computing the total margins).

If the option for a square grid is specified, the right, or top, total margin will be adjusted so the remaining area will be square.

If the total margins and the grid limits (XL, XR, YB, and YT) meet certain error tests, they are used as arguments of XSCALV and YSCALV in computing the scale factors required for generating the grid and plotting on it.

Since so many items influence the margin assignments, the grid boundaries will rarely fall at the exact raster positions that the programmer might have estimated. If the precise raster positions of the boundaries are required in a program, they should be derived by converting the limits into raster counts after GRID1V has been called. For example:

```
IXL= NSV (XL)
IXR= NXV (XR)          (The right total margin is 1023-IXR)
IYB= NYV (YB)
IYT= NYV (YT)          (The top total margin is 1023-IYT)
```

GRID1V ERROR PROCEDURE ERMRKV

Even if bad input data is used, GRID1V will attempt to produce a grid. The philosophy is that some useful information may be revealed, even if the grid is inaccurate.

The grid limits and total margins are tested by lower-level subprograms, ERRLNV (for linear mode) and/or ERRNLV (for log mode). If these tests show that a grid cannot be produced from the given information, some data values are manufactured so that the program can continue. The manufactured quantities are used only internally; data values in the main program will not be affected. (Since ERRLNV and ERRNLV are meant to be used only by GRID1V, the call statements are not given.)

When artificial quantities are used, an error mark (//////////) is placed in the upper right corner of the frame by ERMRKV. Although ERMRKV was designed for use by GRID1V, the programmer can place this mark on the frame if he uses the statement:

CALL ERMRKV

An error mark from GRID1V may indicate that one or more of the following errors has been found:

1. Equal values have been specified for grid limits; that its, XL=XR and/or YB=YT. See Figure 2-16.

2. The specified margins are too wide. In other words, MR+ML $\geq$1023, and/or MB+MT$\geq$1023. See Figure 2-17.

3. In a linear grid, the specified values of DX and/or DY would result in grid lines spaced closer than 3 raster counts.

4. In a log grid the value of one or more of the limits is either zero or negative.

5. In the log mode, there are more than 10 cycles in either the X or Y direction.

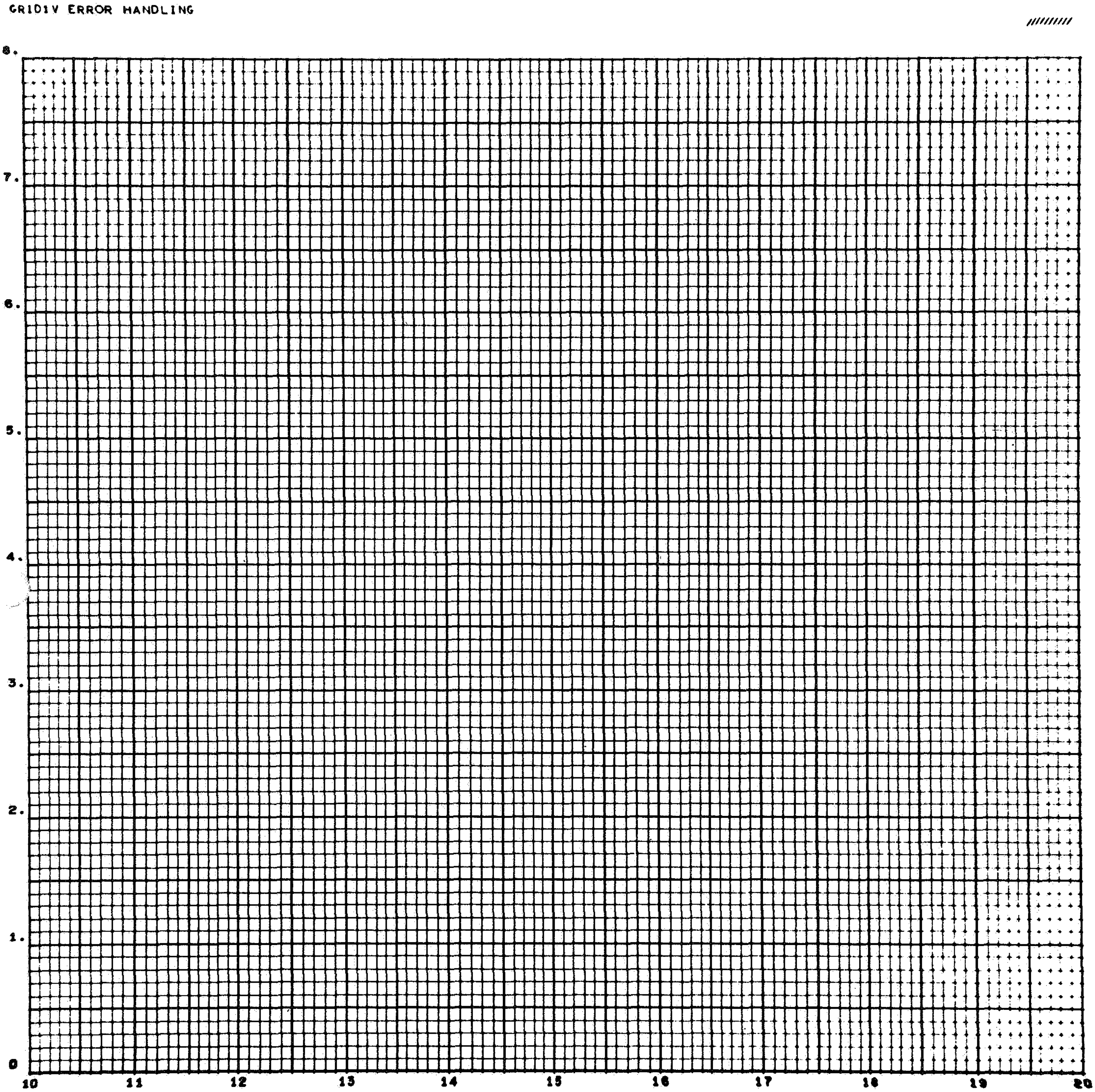
EXAMPLES OF GRID1V CHARACTERISTICS

Certain special characteristics of the grid generated by GRID1V are shown in the following examples.

Example 1. During construction of a linear grid, vertical lines are displayed at intervals "stepped off" from X=0 in DX increment until the maximum limit of the grid is exceeded. They are "stepped off" from X=0 in the negative direction until the minimum limit of the grid is passed. Similarly, horizontal lines are "stepped off" from Y=0 in DY increments. Because of this method of construction, lines will be generated at the grid limits only if XL and XR are integer multiples of DX, and YB and YT are integer multiples of DY. Figures 2-18 and

SPECIAL GRID1V EXAMPLES

GRID1V ERROR HANDLING



CALL GRID1V (1, 10.0, 10.0, 0.0, 8.0, 0.1, 0.1, 5, 5, 10, 10, 2, 2)

Figure 2-16

SPECIAL GRIDIV EXAMPLES

0000-00
001 001

GRIDIV ERROR HANDLING


```
CALL SETHIV ( 600, 600, 40, 40 )
CALL GRIDIV ( 1, 0.0, 10.0, 0.0, 10.0, 1.0, 1.0, 5, 5, 10, 10, 2, 2 )
```

Figure 2-17

2-19 illustrate the effect of changing DX and DY. Figures 2-20 and 2-21 illustrate a similar effect in the negative range of values.

Example 2. Vertical lines to be emphasized are "stepped off" from X=0 by increments of N·DX, and vertical lines to be labeled are "stepped off" by increments of I·DX. Similarly, horizontal lines to be emphasized are "stepped off" from Y=0 by increments of M·DY, and those to be labeled by increments of J·DY. This procedure means that labels and/or emphasized lines will not necessarily occur at the grid limits. Lines at XL and XR will be emphasized only if XI and XR are integer multiples of N·DX. In the same way, lines at YB and YT will be emphasized only if YB and YT are integer multiples of M·DY and labeled only if they are integer multiples of J·DY

This method of determining the positions of line labels is responsible for the absence of labels of the limit lines in Figures 2-19 and 2-21. Notice in the example that the labels for vertical grid lines are adequate, there is no real need to provide labels on the lines at XL and XR. However, the labels for the horizontal lines in this illustration are not adequate; they demonstrate another factor to be considered in planning for line labels. To be meaningful, labels should appear on at least two lines. In the example, the use of J=5, instead of J=10, would cause labeling of the lines of Y=25.0, 30.0, and 35.0, giving a scale that can be read easily.

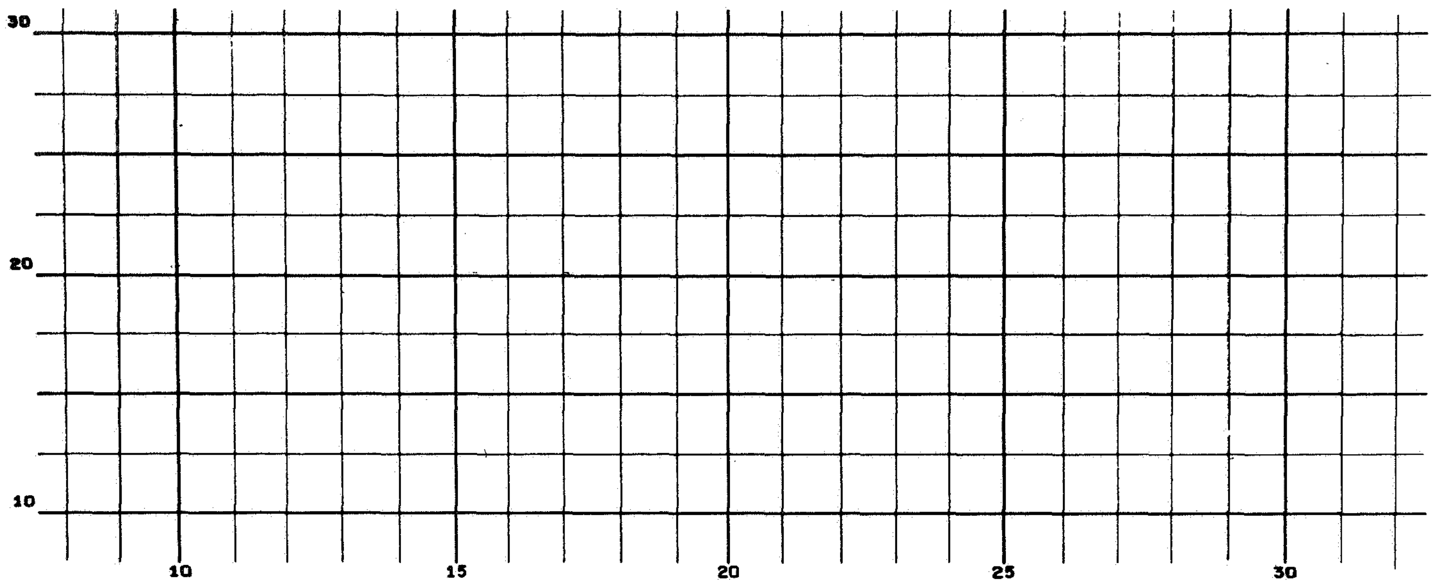
Example 3. Under certain conditions, GRID1V purposely omits labels. Figure 2-22 shows how this can happen. The limits of X used for the grid were -99.9999 and 99.9999, and the limits of Y were 0.0 and 9.9999. During scaling, these limits were rounded, causing them to be treated as -100.0, 100.0, and 0.0, 10.0.

The process caused the decimal scales of the grid limits to be larger than the initial limits, i.e., the values of NX and NY specified by the programmer were

SPECIAL GRID1V EXAMPLES

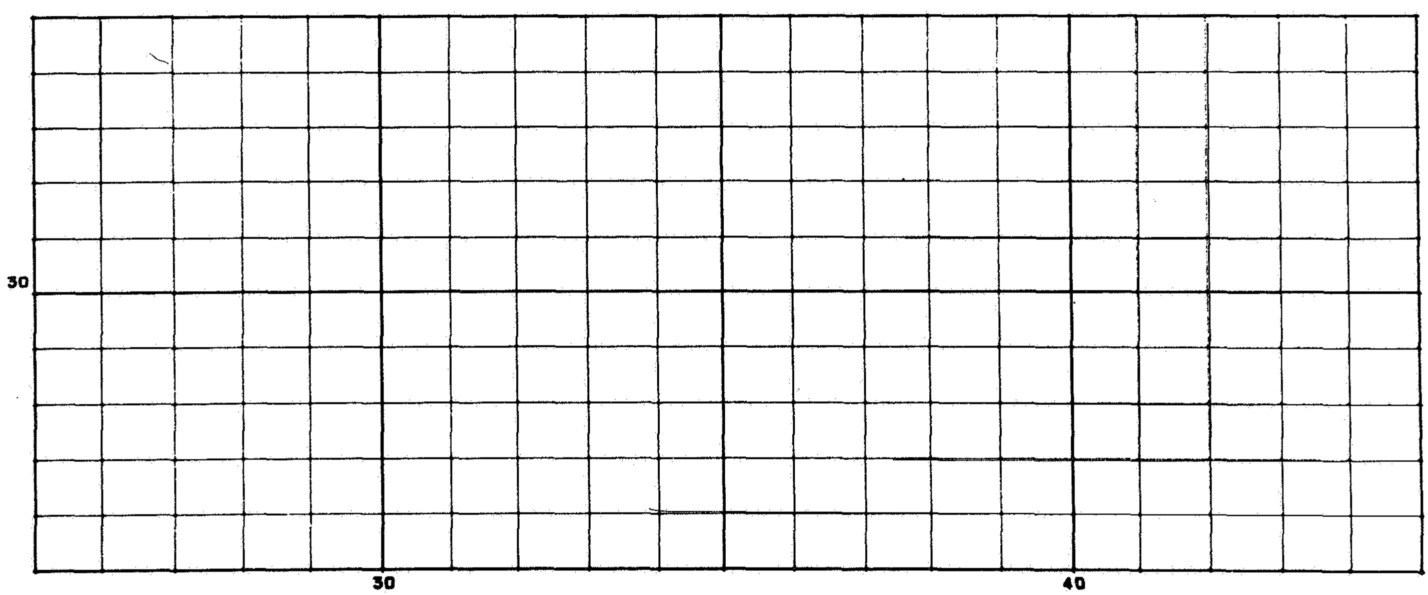
EFFECTS OF STEPPING OFF DELTA INCREMENTS FROM ZERO

0063 0064



CALL GRID1V (1, 7.5, 32.5, 8.0, 31.0, 1.0, 2.5, 5, 2, 5, 4, 2, 2)

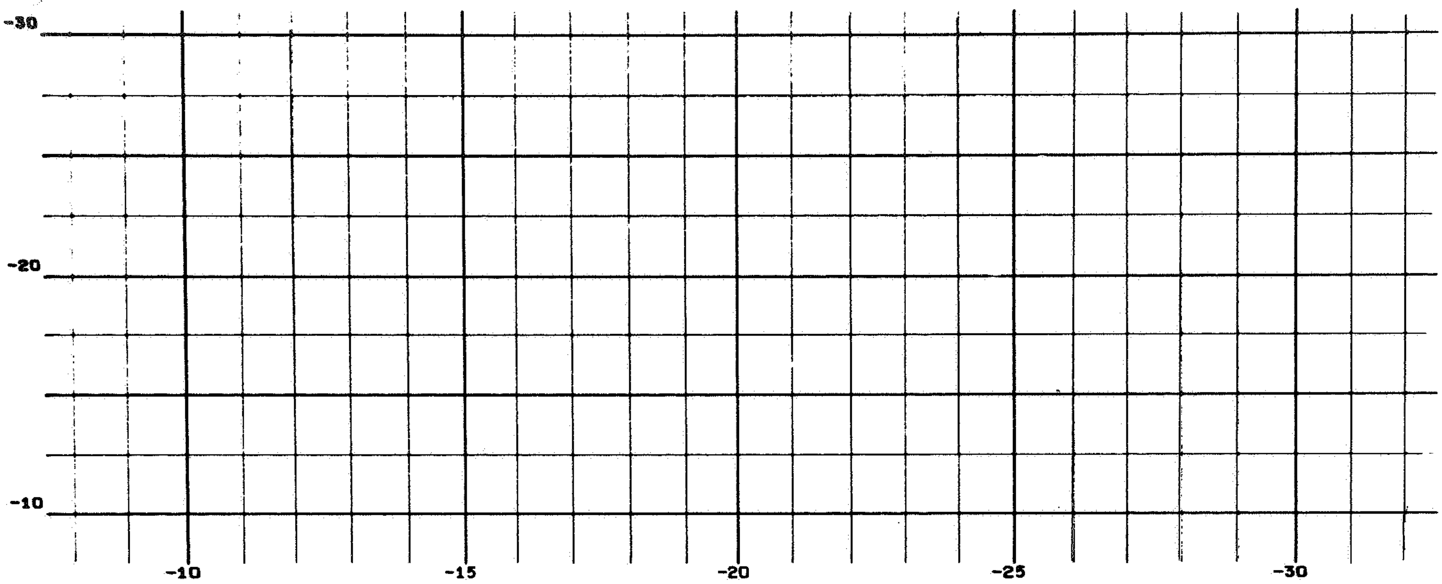
Figure 2-18



CALL GRID1V (2, 25.0, 45.0, 25.0, 35.0, 1.0, 1.0, 5, 5, 10, 10, 2, 2)

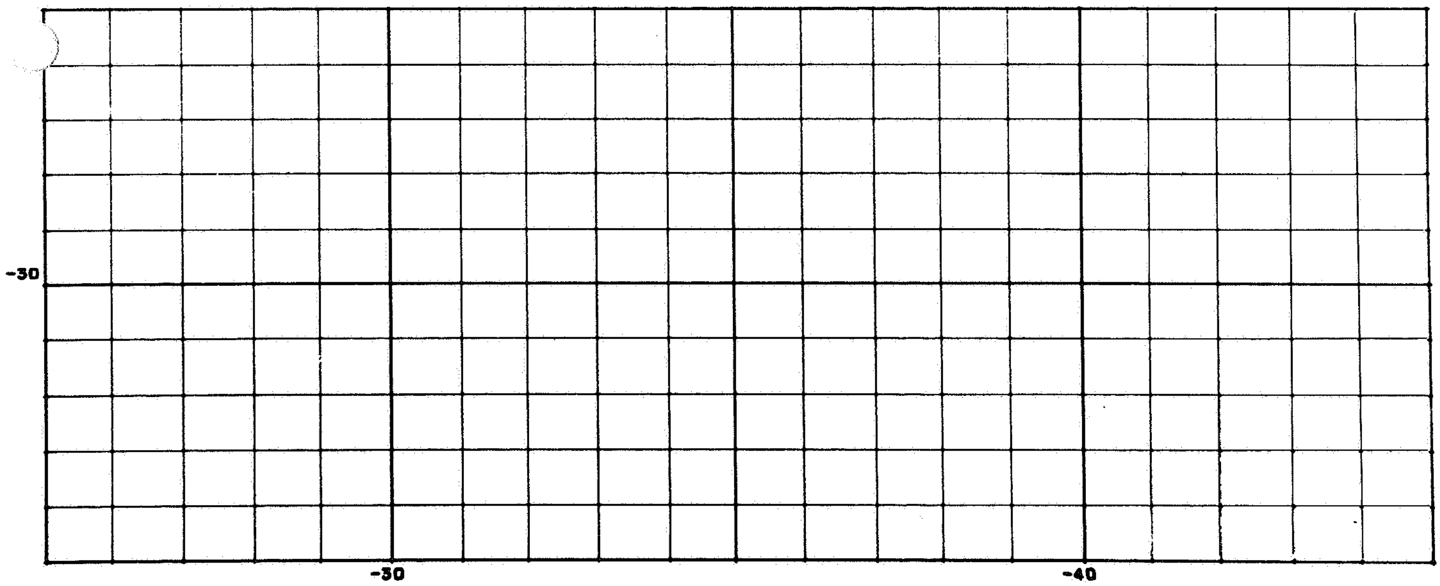
Figure 2-19

THE NEGATIVE RANGE OF VALUES IS HANDLED IN A SIMILAR WAY



CALL GRID1V (1, -7.5, -32.5, -8.0, -31.0, 1.0, 2.5, 5, 2, 5, 4, 2, 2)

Figure 2-20



CALL GRID1V (2, -25.0, -45.0, -25.0, -35.0, 1.0, 1.0, 5, 5, 10, 10, 2, 2)

Figure 2-21

not compatible with the new limits. To avoid erroneous labeling, label values will not be displayed if the rounding process causes the decimal scale to be larger than the initial value.

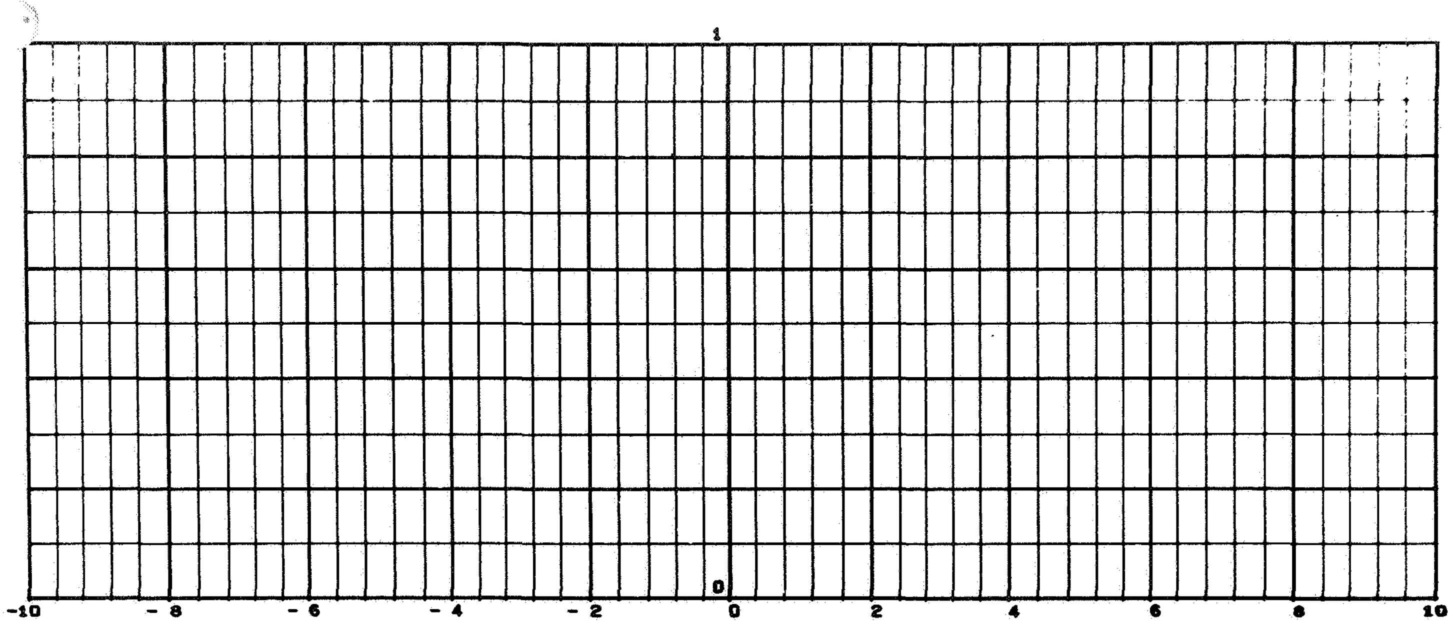
The system does not adequately provide for all problems of this type. Such complications can be avoided if the programmer considers the rounded values of XL and XR, and YB and YT, when specifying NX and NY.

Example 4. When X=0 (and/or Y=0) lies within the grid limits, and I (and/or J) is positive, GRID1V places labels along the X=0 (or Y=0) line. If there isn't enough space for the label between X=0 and the left limit of the grid (or between Y=0 and the bottom of the grid), GRID1V will provide space outside the grid area for labels, just as if negative values of I (and/or J) had been specified. An illustration is shown in Figure 2-23.

Example 5. The number of label characters, NX (or NY), specified for fixed point labels should satisfy the largest and smallest label values to be displayed. Normally, the quantity should be the sum of: (1) the decimal scale of the largest value of X (or Y), (2) the number of fractional positions required in the smallest value, and (3) one more position to provide for the decimal point, if required. The total quantity may not exceed 6 (or 7 if the decimal point is included).

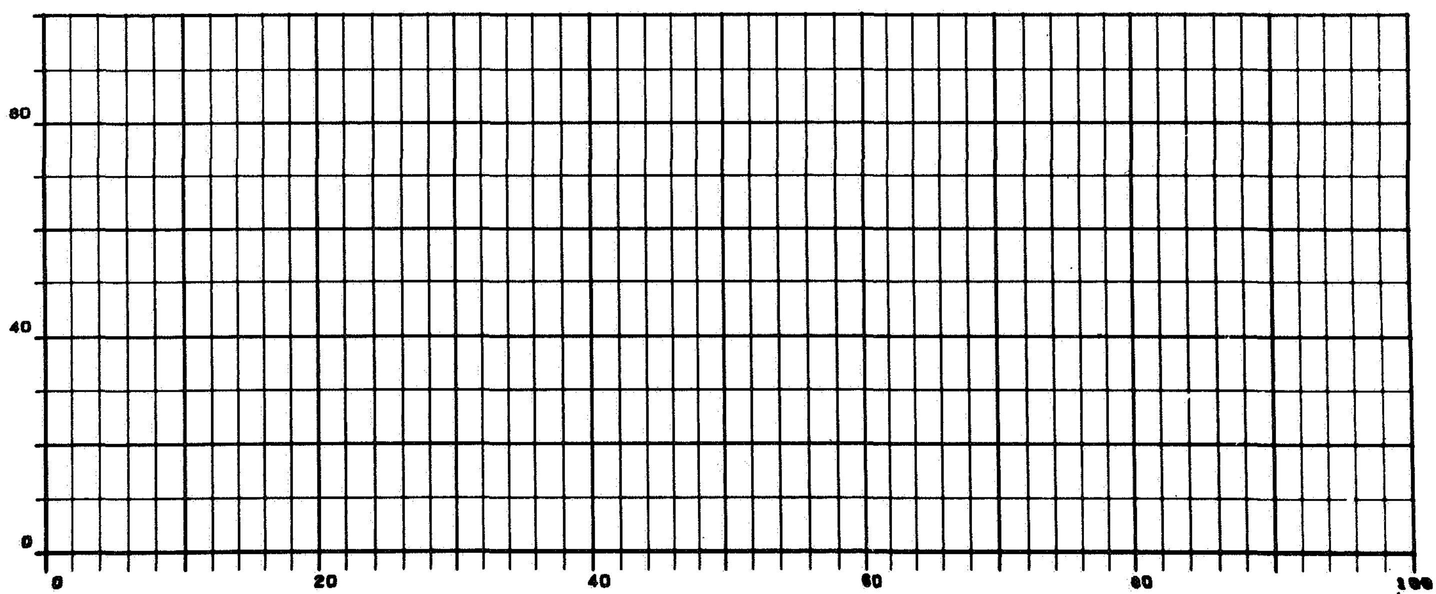
Figure 2-24 illustrates how trouble can occur when NX and/or NY are not specified properly. Note that the values of NX=1 and NY=1 are adequate for the largest values of X and Y, but do not allow for the fractional values required in some of the labels. In this example, 3 should have been used for the values of NX and NY, in the top graph.

SPECIAL CASES AFFECTING GRID LABELS



CALL GRID1V (1, -99.9999 , 99.9999 , 0.0, 9.9999, 4.0, 1.0, 5, 2, 5, 2, 2, 1)

Figure 2-22



CALL GRID1V (2, -0.7, 100.0, -3.0, 100.0, 2.0, 10.0, 5, 2, 10, 4, 3, 3)

Figure 2-23

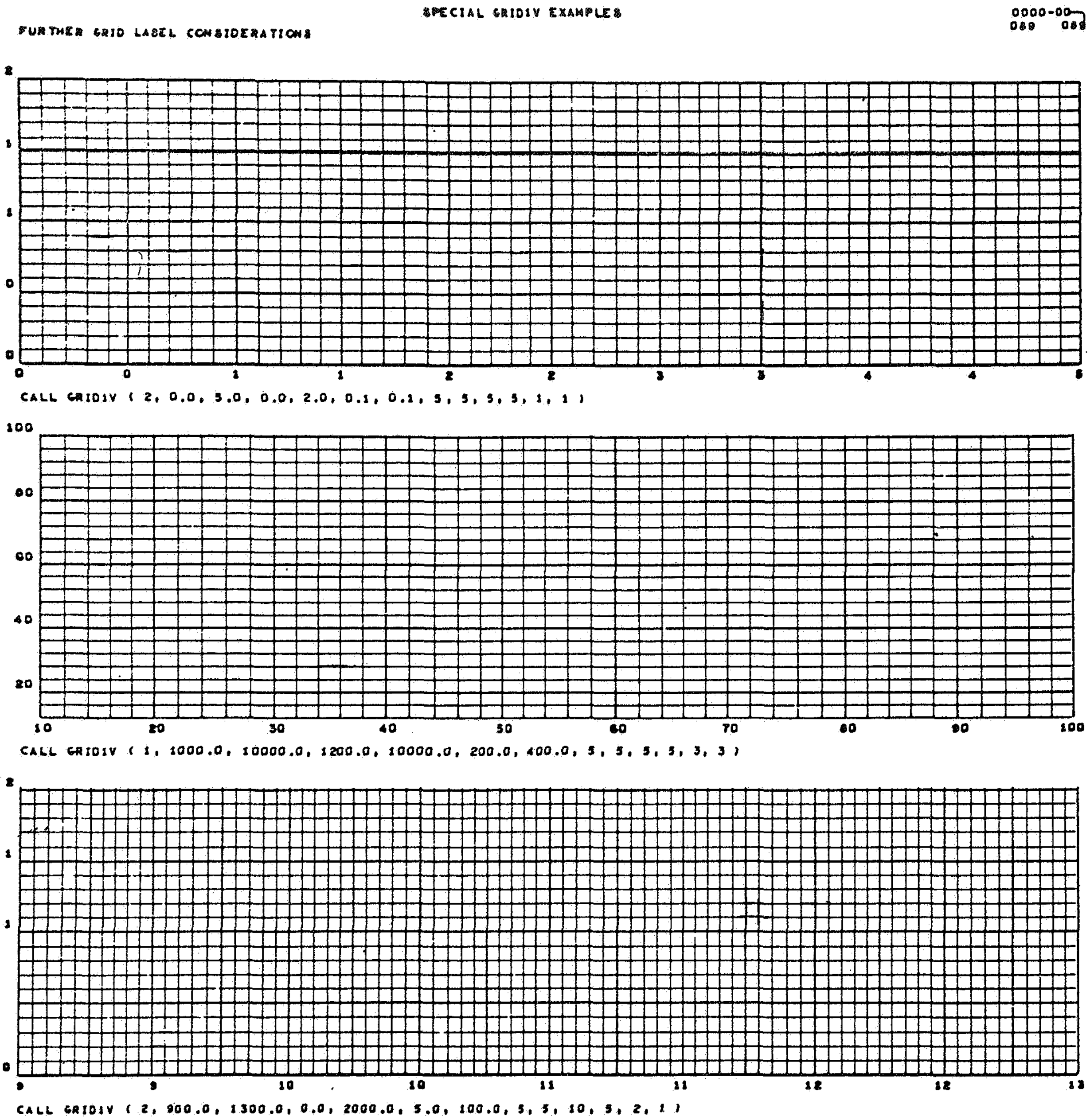

SPECIAL GRIDIV EXAMPLES
FURTHER GRID LABEL CONSIDERATIONS
0000-00
089 089
CALL GRIDIV (2, 0.0, 5.0, 0.0, 2.0, 0.1, 0.1, 5, 5, 5, 5, 1, 1)
CALL GRIDIV (1, 1000.0, 10000.0, 1200.0, 10000.0, 200.0, 400.0, 5, 5, 5, 5, 3, 3)
CALL GRIDIV (2, 900.0, 1300.0, 0.0, 2000.0, 5.0, 100.0, 5, 5, 10, 5, 2, 1)

Figure 2-24

For some applications, it may be advantageous to specify values of NX and/or NY less than the decimal scale of the largest label value. In the middle grid of Figure 2-24, the two low-order positions in the labels have been dropped intentionally, by using values of NX and NY that are less than the decimal scales of the largest values of X and Y. In this case, the resulting graph has been effectively rescaled.

This procedure must be used with care. In the bottom grid, Figure 2-24, the values of NX and NY are so small that the increment between labeled lines is not reflected in the labels that are displayed.

LOG AND SEMI-LOG GRIDS: SMXYV, MSXYV

The earlier examples used only linear scaling and conversion. The modal subroutine SMXYV can be used to alter the scaling mode such that scaling and conversions will be made in the logarithmic mode.

The call statement for establishing the logarithmic mode is:

```
CALL SMXYV (MX, MY)
```

where MX and MY are scale mode indicators that designate whether the logarithmic or linear mode is to be used:

If $MX \neq 0$, $MY \neq 0$, Log in X, Log in Y
$MX \neq 0$, $MY = 0$, Log in X, Linear in Y
$MX = 0$, $MY \neq 0$, Linear in X, Log in Y
$MX = 0$, $MY = 0$, Linear in both X and Y (To restore linear mode)

At the beginning of each job, MX and MY are zero, so that linear scaling and conversion result if SMXYV is never called.

If the programmer wants to generate a log or semi-log plot, he must call SMXYV to set the logarithmic mode before using any S-C 4020 subroutines that involve scaling or conversion.

Once the log-log or semi-log mode has been set by SMXYV and the scale factors have been established, the function statements NXV and NYV can be used to convert data coordinates into raster coordinates, in the same manner as shown for linear scaling.

For <u>chain</u> jobs, note that the values of MX and MY do not carry over from link to link. If a scale mode other than linear-linear is desired, it will be necessary to restate the SMXYV statement in each chain link.

FAP subprograms use two other references in connection with these indicators, causing the names (XXXX) and (YYYY) to appear on the load map.

The contents of the scale mode indicators may be <u>retrieved</u> by using the following statement:

    CALL MSXYV (MXL, MYL)

        The indicator for the X scale mode will be stored in MXL, and for the Y scale mode in MYL.

GRID1V, IXV, IYV, NXV, NYV, XSCALV, and YSCALV all use this statement to determine what scale mode has been selected by the programmer.

<u>RESTRICTIONS ON LOGARITHMIC MODE</u>

In general, once SMXYV has been called, the programmer can use any of the routines to generate a logarithmic display. (One exception: DXDYV should not be used to generate arguments for GRID1V in the direction in which logarithmic scaling is being used.)

However, some of the GRID1V <u>arguments</u> are restricted in the logarithmic mode. Following is a list of the 13 arguments, with notations as to the arguments affected if the mode is logarithmic in the direction affected by each.

```
CALL  GRID1V (L, XL, XR, YB, YT, DX, DY,
              ±N, ±M, ±I, ±J, ±NX, ±NY)
```

L	Controls film advance.
XL, XR	Left and right limits of the grid. May not be negative or zero.
YB, YT	Bottom and top limits of the grid. May not be negative or zero.
DX, DY	Should be set to 1.0. If less, only the cycle lines will be displayed. If greater, no line will be drawn.
N, M	Will be ignored; however, an argument must be present. A negative sign on N or M will force the grid to the square.
I, J	Will be ignored; however, an argument must be present.
NX, NY	Number of characters to be displayed in the labels.

Generally, line labels will be placed only at the cycle lines. If, however, the grid spans less than one complete cycle, each grid line will be labeled.

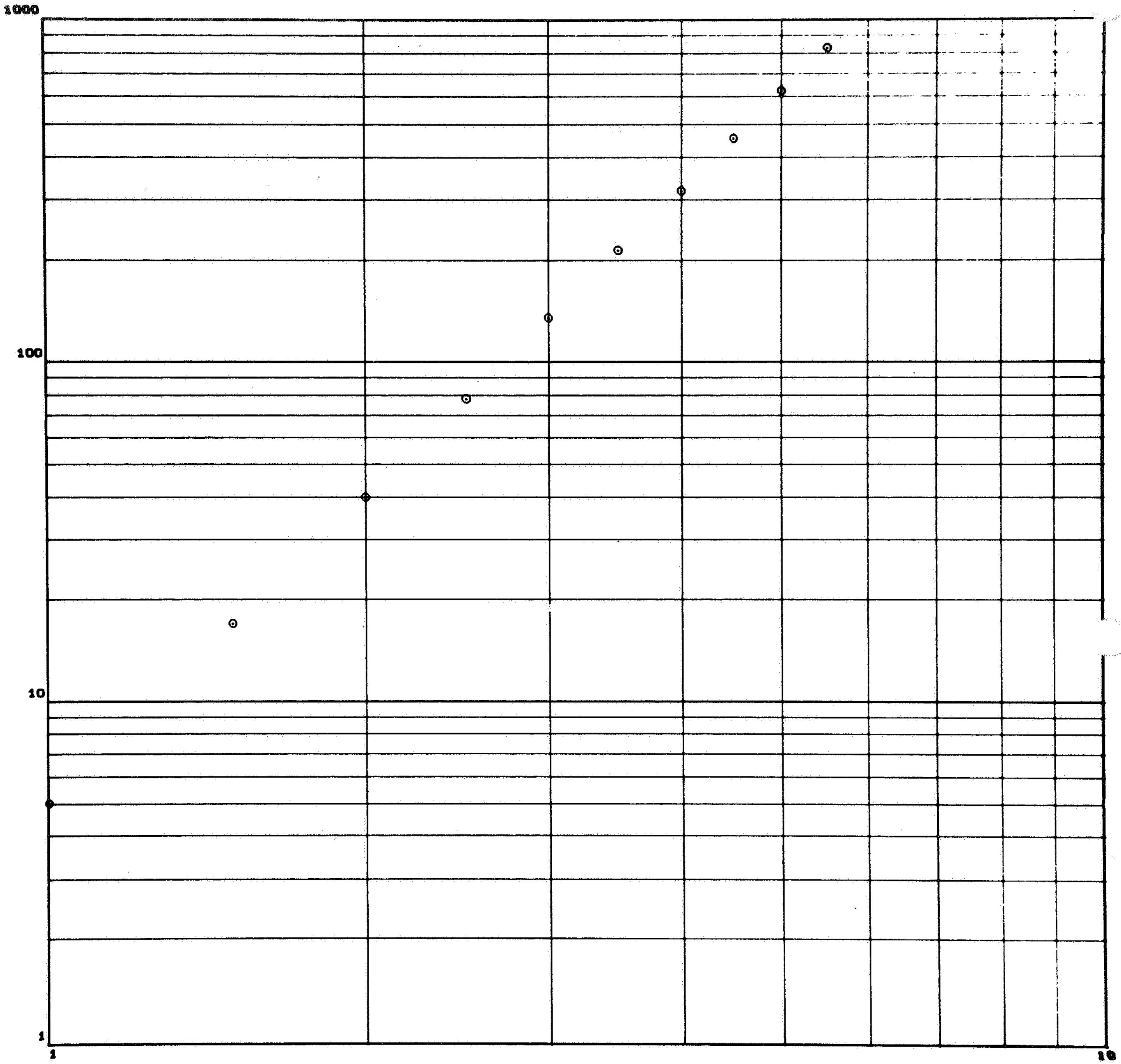
No more than 10 log cycles are permitted in the GRID1V system.

Examples. Figure 2-25 illustrates a plot that is logarithmic in both the X and Y directions. Note the arguments for SMXYV and GRID1V in the coding.

Figure 2-26 illustrates a semi-log grid, with plotting of the data used in Figure 2-25.

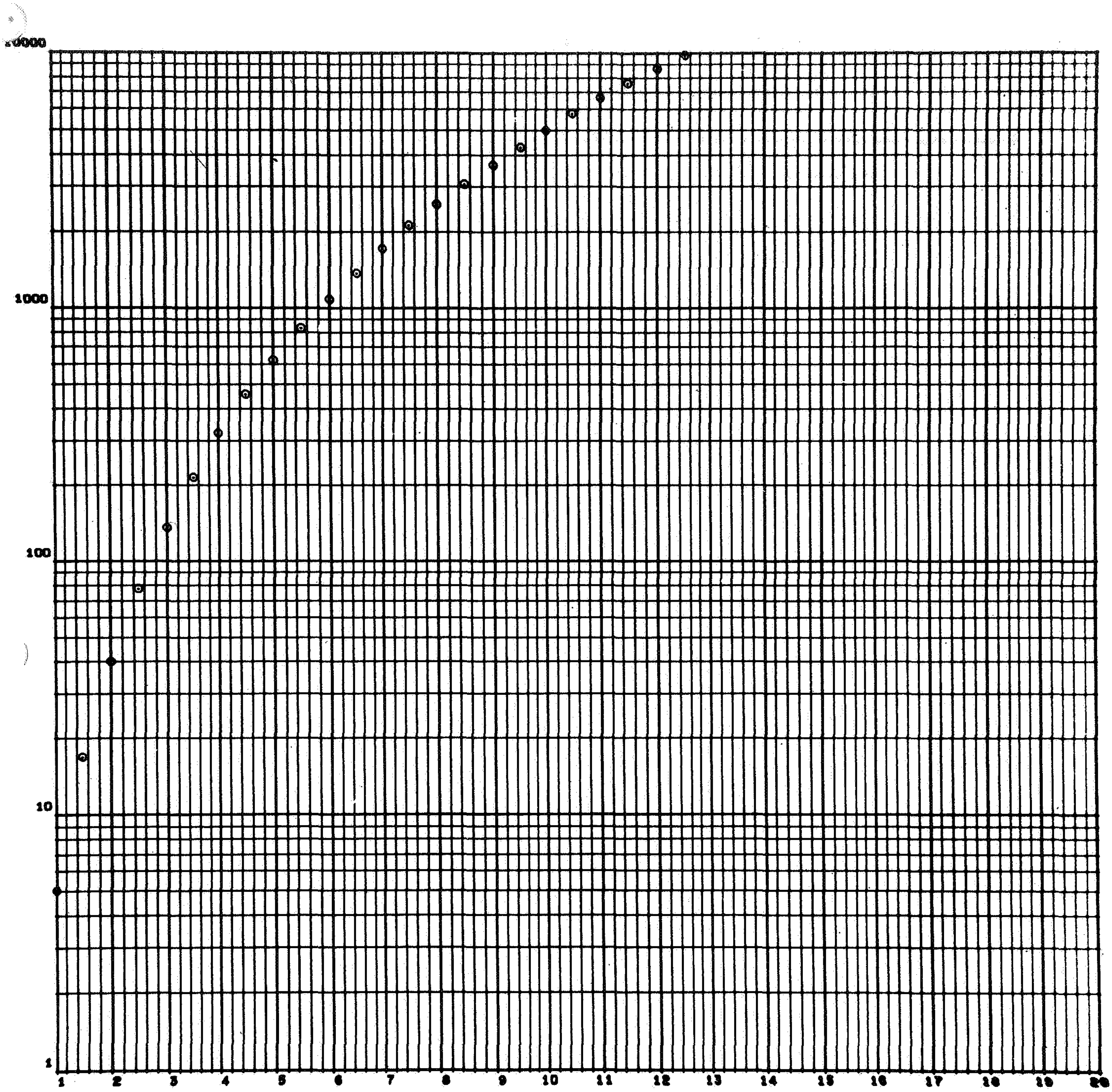
BUILDING SPECIAL GRIDS: LINRV, NONLNV

Two of the lower-level modules employed by GRID1V are useful as building blocks for building special grids. LINRV may be used to generate only the vertical portion or only the horizontal portion of a linear grid. NONLNV can generate only the vertical or only the horizontal portion of a log grid. See examples in Figure 2-27 and Figure 2-28.




```
CALL SMXYV ( 1, 1 )
CALL GRID1V ( 1, 1.0, 10.0, 1.0, 1000.0, 1.0, 1.0, 1, 1, 1, 1, 2, 4 )
        POINTS PLOTTED BY USING CALL POINTV ( X,Y, 1  )  IN A LOOP
```

Figure 2-25




```
CALL SMXYV ( 0, 1 )
CALL DXDYV ( 1, 1.0, 20.0, DX, N, I, NX, 12.0, IERR )
CALL GRID1V ( 1, 1.0, 20.0, 1.0, 10000.0, DX, 1.0, N, 1, I, 1, NX, 5 )
```

Figure 2-26

In addition to generating the vertical and horizontal portions of a grid in separate operations, these modules offer other special capabilities:

1. The programmer can control the length of the grid lines. For example, the grid lines can be as short as "time ticks."

2. The programmer can exert greater control over the position of line labels by specifying a label reference location.

3. Selected grid lines can be emphasized and/or labeled in the same way as with GRID1V.

4. DXDYV can be used with LINRV in much the same way it was used with GRID1V.

Certain subprograms must be executed prior to the use of LINRV or NONLNV. The frame must have been advanced, and XSCALV and/or YSCALV must have established scale factors. If a change in scale mode is required, SMXYV must have been called.

The call statement for using LINRV to generate a vertical grid is:

```
CALL LINRV  (1, LYREFR, IYMIN, IYMAX, XL, XR,
             DX, ± N, ±I, ±NY, IH)
```

For a horizontal grid the statement is:

```
CALL LINRV  (2, LXREFR, IXMIN, IXMAX, YB, YT,
             DY,± M,±J, ±NY, IH)
```

LYREFR	Label references locations.
LXREFR	For a vertical grid, LYREFR is the Y raster coordinate that will be used to position the labels of the vertical grid lines. The X raster coordinate will vary, and will be computed by the subprogram. For a horizontal grid, LXREFR is the X raster coordinate that will be used to position the labels of the horizontal grid lines. The Y raster coordinates will vary, and will be computed by the subprogram. The character dimensions, IW and IH (below), should be considered in assigning LYREFR and LXREFR.

	For example, LYREFR should be chosen to allow at least one character height (IH) below the origin of vertical grid lines (IYMIN). LXREFR should be a raster position that is at least IW* (NY+1) raster counts to the left of the origin of the horizontal grid lines (IXMIN). In both instances, some additional space should be added to prevent overlapping. If scientific labels are selected (-NX and/or -NY), provision must be made for 7 additional label characters in the computation of LXREFR. About 5 raster counts extra must be added to LYREFR to allow for the raised exponent. NOTE: IW* (NY+1) allows space for a sign
IYMIN, IYMAX IXMIN, IXMAX	Raster positions that determine the origin and end of each line. For a vertical grid, these integers are raster counts in the Y direction. For a horizontal grid, these integers are raster counts in the X direction.
IW, IH	The dimensions to be allowed for each label character; IW for the width, and IH for the height. Since LINRV uses LABLV to display the labels, and since the system LABLV uses Charactron characters, the dimensions should be IW=8 and IH=10. (If the standard values have not been altered, these values can be obtained by calling SETCOV.)

The remainder of the arguments have the same definitions given for these terms in the discussion of GRID1V.

The vertical portion of a log grid can be generated by the statement:

```
CALL NONLNV (1, LYREFR, IYMIN, IYMAX, XL, XR, DX,
             ±N, ±I, ±NX, IW)
```

For a horizontal log grid, the statement is:

```
CALL NONLNV (2, LXREFR, IXMIN, IXMAX, YB, YT, DY
             ±M, ±J, ±NY, IH)
```

The arguments XL, XR, DX, N, I, NX, and YB, YT, DY, M, J, NY have the same definitions (and restrictions) given for these terms in the discussion of GRID1V for the log mode. The remaining arguments have the definitions given for LINRV.

Figures 2-27 and 2-28 show examples of portions of grids created by LINRV and NONLNV, respectively.

AXIS LINES: XAXISV, YAXISV

Axis lines XAXISV, YAXISV enable the programmer to use the axis line feature of the S-C 4020 to generate horizontal or vertical lines. Horizontal axis lines started at a specified raster position will be swept to the right to the specified X stop point in raster count. Vertical axis lines started at a specified raster position will be swept upwards to the specified Y stop point in raster count. The axis generator will not sweep lines down or to the left; therefore, the program automatically interchanges the coordinates if it is required in order not to stop the S-C 4020. If no stop point is given in the parameter list, the axis will be swept to the right edge of the frame or the top edge of the frame.

The call statement for sweeping a horizontal line is:

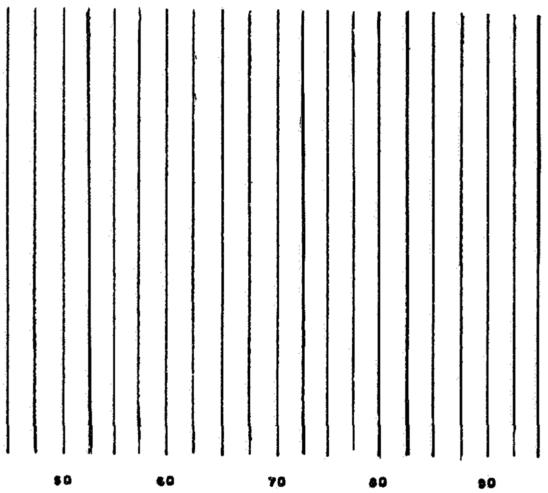
```
      CALL  XAXISV  (IX, IY)

or    CALL  XAXISV  (IX, IY, NSTPT)
```

IX, IY — The fixed point raster coordinates of the origin of the line. IX may have a value from 0 to 959, while IY may have a value from 0 to 959.

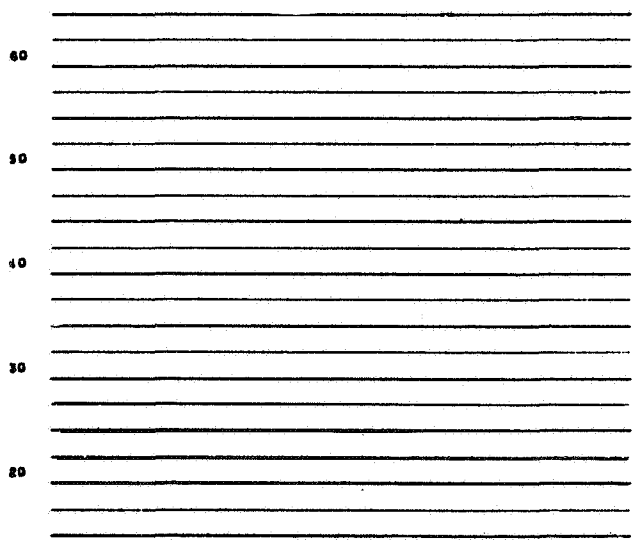
NSTPT — The fixed point X coordinate of the stop point.

For a vertical line, the statement is:

```
      CALL YAXISV  (IX, IY)
```




```
CALL FRAMEV
CALL XSCALV ( 45.0, 95.0, 50, 544 )
CALL LINRV ( 1, 515, 535, 900, 45.0, 95.0, 2.5, 2, 4, 2, 0 )
```




```
CALL YSCALV ( 15.0, 65.0, 20, 575 )
CALL LINRV ( 2, 520, 550, 1023, 15.0, 65.0, 2.5, 2, 4, 2, 12 )
```

Figure 2-27

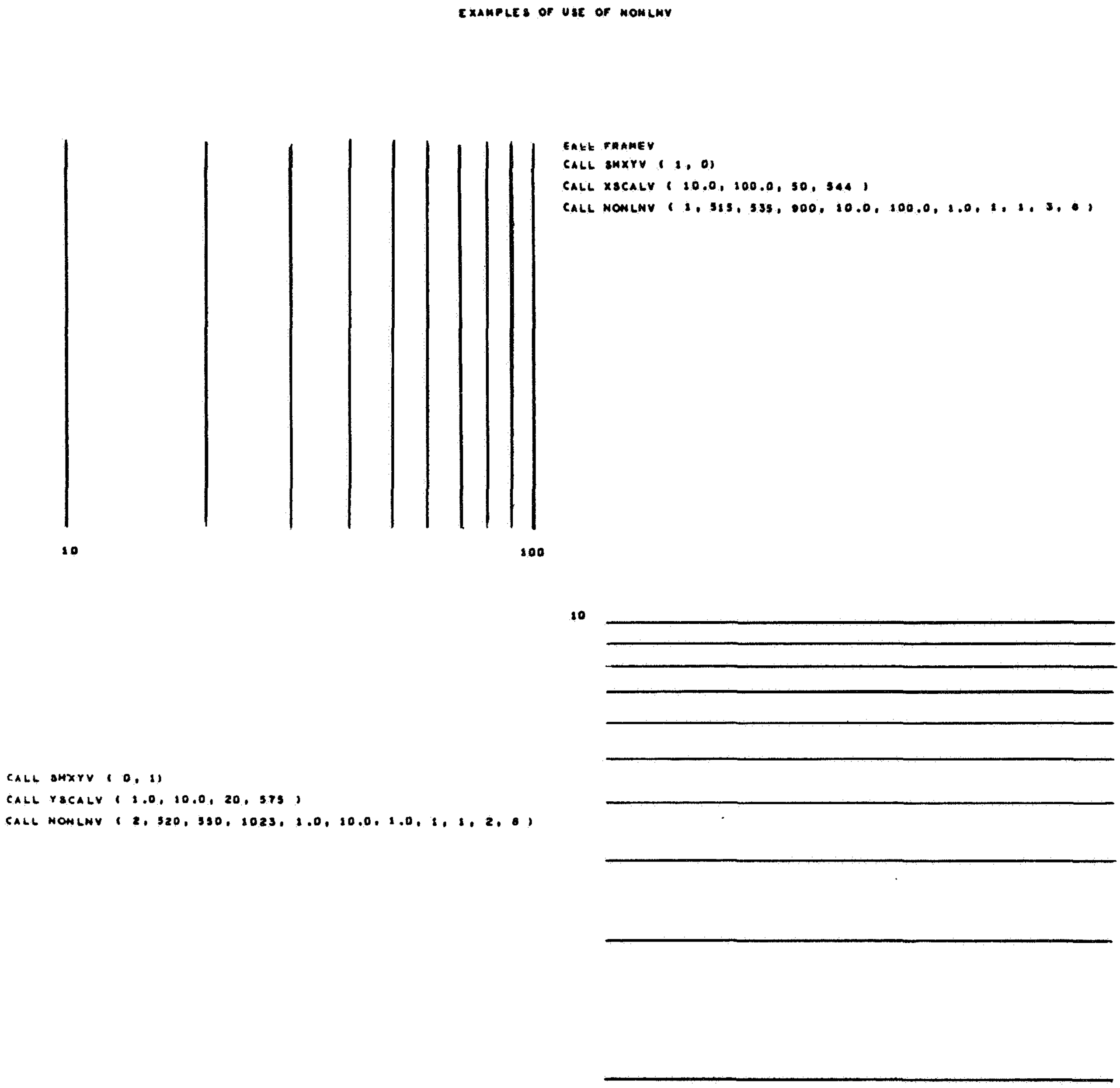

EXAMPLES OF USE OF NONLNV
CALL FRAMEV
CALL SHXYV (1, 0)
CALL XSCALV (10.0, 100.0, 50, 544)
CALL NONLNV (1, 515, 535, 900, 10.0, 100.0, 1.0, 1, 1, 3, 0)
10
100
10
CALL SHXYV (0, 1)
CALL YSCALV (1.0, 10.0, 20, 575)
CALL NONLNV (2, 520, 550, 1023, 1.0, 10.0, 1.0, 1, 1, 2, 8)

Figure 2-28

or CALL YAXISV (IX, IY, NSTPT)

IX, IY	The fixed raster coordinates of the origin of the line. IX may have a value from 0 to 959, while IY may have a value from 0 to 959.
NSTPT	The fixed point Y coordinate of the stop point.

D. SCALING AND CONVERSION

In earlier sections, scaling and conversion problems have been left to the routine GRID1V. However, much of the actual computation is done in lower-level modules. This section describes these and some associated modules that provide additional tools for some scaling and conversion problems.

The descriptions may also be useful in clarifying the operation of higher-level routines. For example, GRID1V uses XSCALV, YSCALV as a lower-level routine to do scaling. Consequently, the comments in this section concerning scaling and conversion equations, scale factors, and retrieving and resetting scale factors also apply when the programmer uses GRID1V to control scaling.

Methods of operation in both the linear and nonlinear modes are discussed. The nonlinear mode built into the system is the logarithmic mode, but the possibility of substitution of other nonlinear modes is mentioned.

BASIC SCALING SUBPROGRAMS: XSCALV, YSCALV

XSCALV, YSCALV will compute the scale factors for a specified display and store them in an internal table for later use by those functions which convert data. The calling statements are:

CALL XSCALV (XL, XR, ML, MR)

CALL YSCALV (YB, YT, MB, MT)

XL, XR — Floating point values of X for the leftmost and rightmost limits of the scaled plotting area.

ML, MR — The amount of margin space to be reserved to the left and right of the scaled area, expressed in raster counts (fixed point integers).

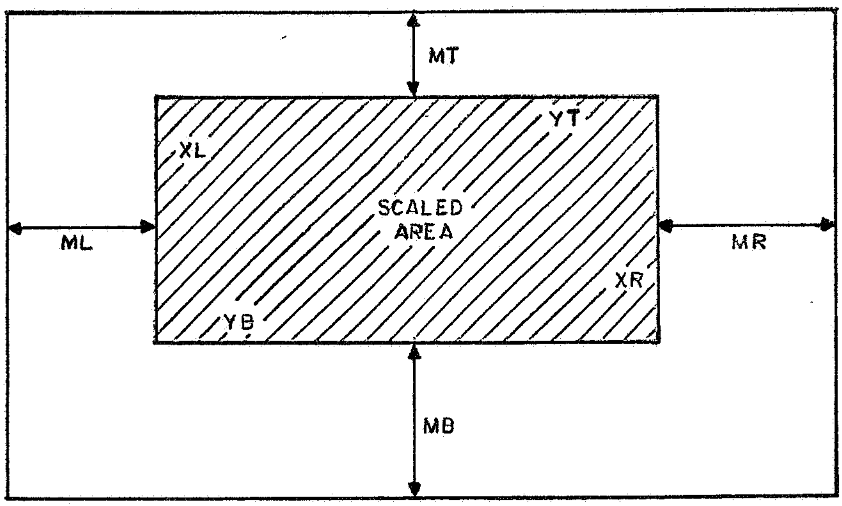
YB, YT — Floating point values of Y for the bottom and top limits of the scaled plotting area.

MB, MT — The amount of margin space to be reserved below and above the scaled area, expressed in raster counts (fixed point integers).

XSCALV, YSCALV contains a test for nonlinear mode. If this mode is indicated, XL, XR, and/or YB, YT will be transformed before the scale factors are computed by the basic scaling equations.

<u>Example</u>

Figure 2-29 illustrates the relationship of the arguments. The margin specifications are: ML=170, MR=192, MB=340, MT=128.



XSCALV will assign XL to raster location IX=170, and XR to raster location IX=831 (i.e., 1023-192). YSCALV will assign YB to raster location IY-340, and YT to raster location IY=895 (i.e., 1023-128). The scaled area will then be the rectangle from IX=170 to IX=831, and from IY=340 to IY=895.

BASIC SCALING EQUATIONS

In the following equations, "A" and "B" represent the scale factors computed and stored by XSCALV, and "C" and "D" are the factors computed and stored by YSCALV. (Since the computation is done in floating point arithmetic, the floating point variable names FML, FMR, and FMT are used to represent the floating point equivalents of the margin values ML, MR, MB, and MT.)

$$A = \frac{(1023. - FMR) - FML}{XR - XL} \qquad \text{I}$$

$$B = FML - A*XL \qquad \text{II}$$

$$C = \frac{(1023. - FMT) - FMB}{YT - YB} \qquad \text{III}$$

$$D = FMB - C*YB \qquad \text{IV}$$

CONVERSION OF DATA: NXV, NYV, IXV, IYV

Four function subprograms, NXV, NYV, IXV, and IYV, are provided to convert data coordinates into raster coordinates. The argument for each of the functions must be a floating point quantity; the result will be an integer quantity.

The following FORTRAN statements show how these functions may be used to convert data coordinates X (or Y) into raster coordinates IX (or IY):

```
IX= NXV(X)
IY= NYV(Y)
IX= IXV(X)
IY= IYV(Y)
```

These four functions are <u>similar</u> in that they all convert data by means of the basic equations for data conversion discussed below. They are <u>dissimilar</u> in the way they handle off-scale data (that is, data which falls outside the limits XL, XR, or YB, YT.)

The functions NXV and NYV check for off-scale data values. The result IX (or IY) will be set to zero if the argument X (or Y) is outside the limits that were used to establish the scale. In addition, an error indication is set, as discussed under Off-Scale Error Detection.

The functions IXV and IYV do not test for off-scale data values. The resulting position can be outside the plotting area, or even outside the frame, but the value will be properly scaled relative to the plotting area. (However, no test is made for the possibility that the result is greater than 131,071; integer bits above the 17th will be lost.)

Figure 2-30 illustrates how error testing of the results of NXV, NYV can be used to bypass plotting of points that are off-scale. NXV and NYV were used to convert the points along the curve into raster positions, and LINEV was employed to connect the points. Since the results of NXV and NYV were tested for zeros, and plotting was by-passed whenever a point was off-scale, the curve stopped at the top and right limits of the scaled area (outlines).

In Figure 2-31 two sine curves are shown, one plotted after using NXV, NYV to do the conversion, and the other after IXV, IYV were used. As in Figure 2-30, LINEV was used to connect the points. NO ERROR TESTS WERE MADE.

The lower curve shows the line going to zero when off-scale values were encountered by NXV, NYV. Note that this curve drops to the bottom of the frame (IY=0) when values that were off-scale in Y were encountered. Also note that the off-scale initial and last values of X caused the curve to start and end at the left edge of the frame (IX=0).

The higher curve was drawn after IXV, IYV were used to convert the points. The curve continued past the boundaries of the scaled area when off-scale values were

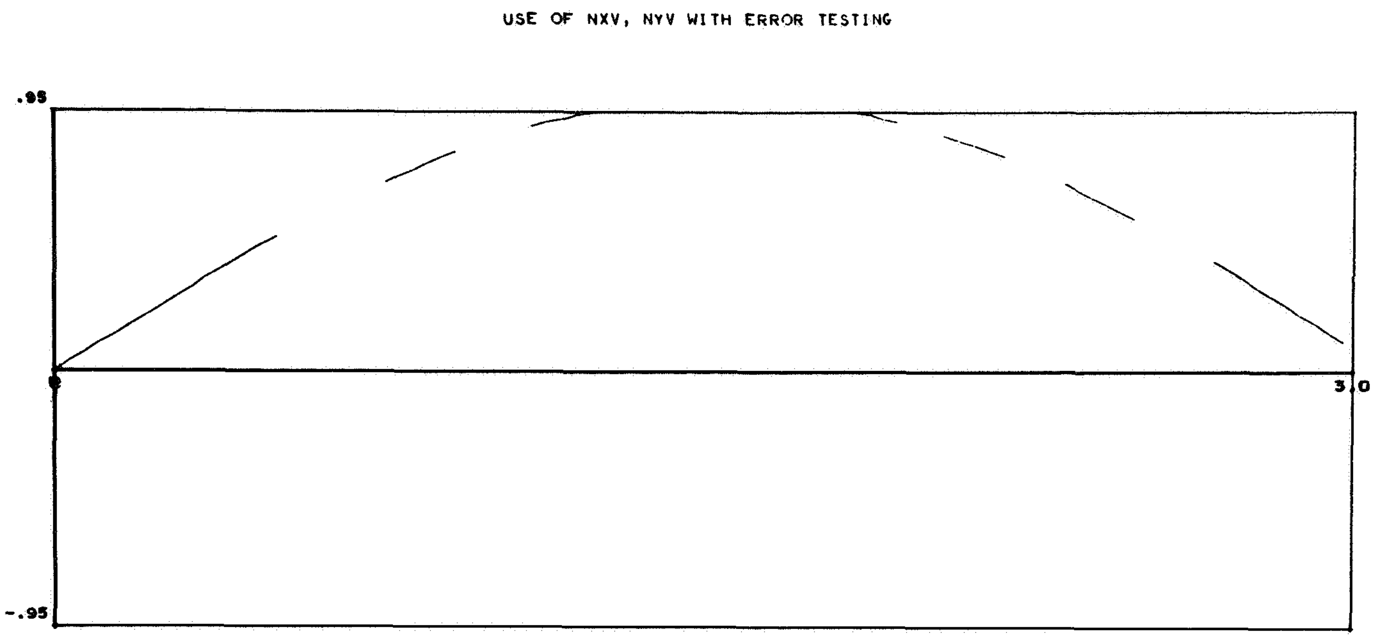


Figure 2-30

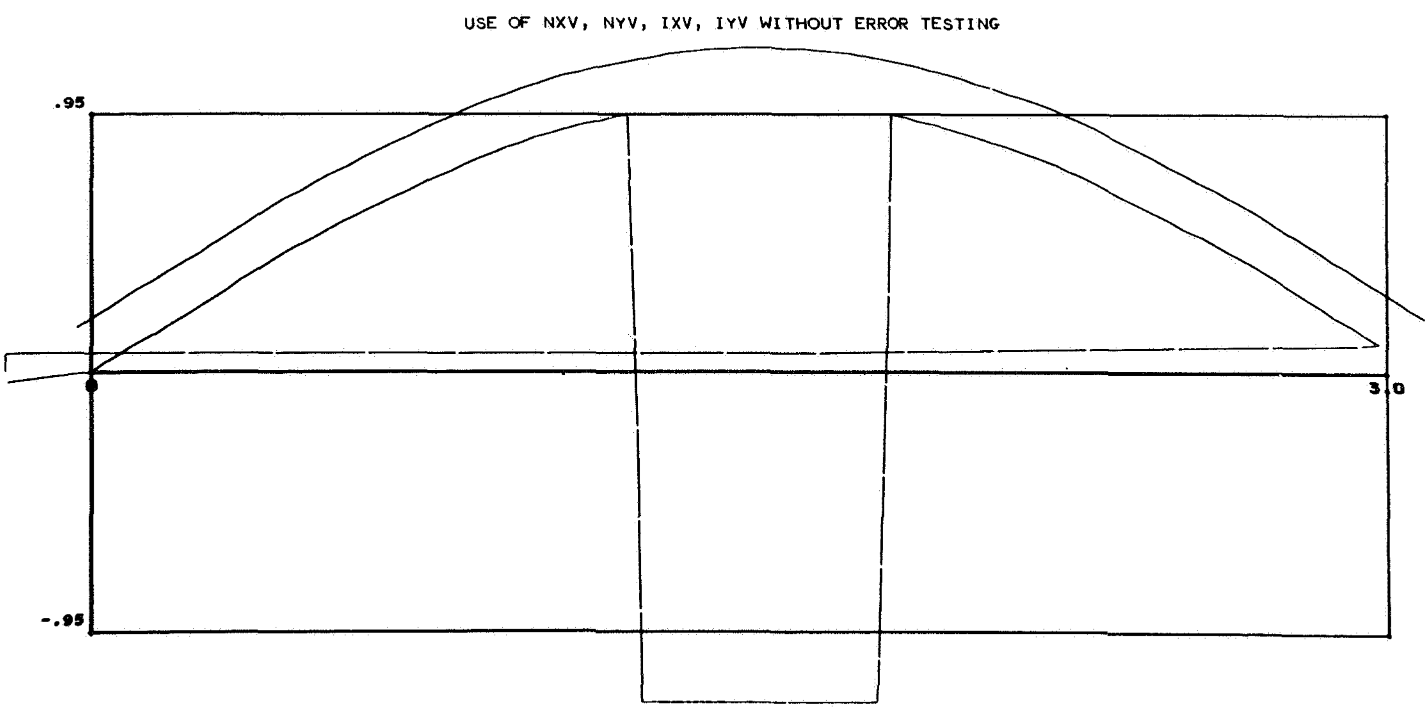


Figure 2-31

encountered. (When IXV, IYV are used, the programmer must decide what action should be taken when the result of IXV and/or IYV is < 0 or > 1023.)

BASIC CONVERSION EQUATIONS

The following equations show how the conversion functions convert data coordinates X and Y into raster coordinates IX and IY:

IX= A*X + B

IY= C*Y + D

The scale factors A, B, C, and D are those derived from the equations I, II, III, and IV.

Generally speaking, the programmer should use the conversion functions rather than writing statements of his own containing these equations. The functions offer the following advantages:

a. They have direct access to the internal table in which the scale A, B, C, and D are stored.

b. They check the scale mode, and use a nonlinear conversion if that mode is indicated.

c. NXV, NYV contain a test for off-scale points.

If it is ever necessary to use these equations directly, the programmer can retrieve the scale factors (A, B, C, D) by employing the routine SCLSAV.

INVERSE CONVERSION UXV, UYV

The functions UXV, UYV allow the programmer to obtain the coordinates of a specified raster location in terms of his data. The following statements show how these functions may be used:

X= UXV (IX)

Y= UYV (IY)

Although UXV, UYV represent the inverse of IX= IXV (X) and IY= IYV (Y), the results are approximate because truncation occurs in the IXV and IYV functions.

NOTE

UXV, UYV CANNOT BE USED IN THE NONLINEAR MODE.

The equations used by UXV, UYV are the inverse of the equations for data conversion:

$$X = \frac{IX - B}{A}$$

$$Y = \frac{IY - D}{C}$$

(The computation is performed in floating point arithmetic.)

RETRIEVAL OF SCALING INFORMATION: SCLSAV

The subroutine SCLSAV will retrieve scale factors and other scaling information from an internal table and store them in an array named by the programmer. Although SCLSAV makes scaling information available for special-purpose conversions, certain limit tests, etc., its principal value is that it permits saving scaling information from one program link to another. When a new link is entered, another routine RESCLV, can be called to restore the scaling information in the internal table, where it is accessible to NXV, NYV, IXV, or IYV.

The calling statement to retrieve the scaling information is:

CALL SCLSAV (R)

R	The name of a ten-cell array, named and dimensioned by the programmer.

The storage locations in the block of cells, R, are assigned as follows:

R(10)	Minimum IY
R(9)	Minimum IX
R(8)	Maximum IY

R(7)	Maximum IX	
R(6)	D	
R(5)	B	Scale factors
R(4)	C	
R(3)	A	
R(2)	Scale mode indicator for Y	
R(1)	Scale mode indicator for X	

RESETTING SCALING INFORMATION: RESCLV

If the scaling information has been stored in COMMON by SCLSAV, it can be reset into the internal table when a new chain link is entered. The statement is:

```
CALL RESCLV (R)
```

where R is the ten-cell array described under SCLSAV.

NONLINEAR SCALING AND CONVERSION

The only nonlinear capability built into the system is logarithmic scale mode, used in connection with log grids. However, the system design allows the programmer to incorporate some special nonlinear scale mode, by the substitution of one module of his own. For this reason, the more inclusive term, "nonlinear", is used in the following discussion, instead of "logarithmic." The linear and nonlinear scale mode indicators are discussed on page number II-47.

NONLINEAR TRANSFORMATION: XMODV, YMODV

The scaling and conversion equations shown under number conversion apply not only to the linear mode, but also to the nonlinear mode <u>if transformed arguments are used.</u> The functions XMODV, YMODV are provided to perform such transformation; they can be used in statements of the following type:

```
XPRIME = XMODV (X)
```

YPRIME = YMODV (Y)

The scaling and conversion subprograms (XSCALV, YSCALV, IXV, IYV, NXV, NYV) test the scale mode indicators to see if they have been set to nonlinear mode by a call SMXYV. If nonlinear mode is indicated, each of these subprograms will use XMODV (or YMODV) to perform a nonlinear transformation on X (or Y) before the remainder of the scaling or conversion takes place. Since the XMODV, YMODV functions in the system compute the log of X or Y, the system nonlinear mode is synonymous with log mode.

For a problem requiring special nonlinear transformation, the programmer can substitute subprograms of his own named XMODV and YMODV, with one argument and one result. Since the system XMODV and YMODV are physically contained in one subprogram, both must be replaced if a substitution is made for either. Obviously, such substitute functions must meet the nonlinear scaling and conversion requirements of the entire program (or chain link), since they will replace the system functions.

System subprograms that use XMODV and YMODV (either directly or indirectly) are: GRID1V, IXV, IYV, LINRV, NONLNV, NXV, NYV, XSCALV, YSCALV, APLOTV, POINTV.

WARNING: GRID1V WAS DESIGNED ONLY FOR THE LINEAR, LOG, AND SEMI-LOG OPTIONS. SUBSTITUTION OF A SPECIAL TRANSFORMATION FUNCTION MAY CAUSE UNEXPECTED DIFFICULTIES IF USED BY GRID1V.

OFF-SCALE ERROR DETECTION

Whenever there is a possibility that off-scale data points might be encountered, error tests should be made by the programmer. For example, APLOTV sets an error indicator which should be tested. A zero result from NXV or NYV nearly always indicates an error; a test should be made for this condition.

There are additional situations which demand special error detection procedures. For one thing, if no left and/or bottom margin space is reserved, the conversion of XL and/or YB can produce a legitimate zero result from NXV, NYV. More important, the programmer may be using NXV or NYV indirectly, via other modules, and thus be unable to test the results. For these reasons, additional subprograms are provided for detailed analysis of conversion errors resulting from off-scale points.

Keep in mind that the special procedures which follow are designed for unusual situations, in which normal error testing does not suffice.

SET CONVERSION ERROR INDICATORS: SCERRV

Two internal cells are used by NXV and NYV to store indications of successful or unsuccessful data conversion. The subroutine SCERRV allows the programmer to assign two cells which NXV, NYV will use in place of the internal error cells. In this way, the programmer can name error indicator locations that are accessible to his program. The call statement is:

```
CALL SCERRV (KX , KY)
```

When NXV converts a quantity successfully, it will place a "0" in KX; when unsuccessful, it will store a "1" in KX. NYV will use the cell KY in the same way. The cells named will be used in each subsequent execution of NXV, NYV (including execution via other subprograms) until new cells are named by another call to SCERRV (or the internal cells are reset at the beginning of a new link of a chain job.) If possible, tests of these error cells should be made as soon after execution of NXV, NYV as possible, to avoid any possibility that they might be altered by subsequent executions of NXV, NYV.

The following practical examples show how SCERRV can provide error indicator cells in connection with POINTV. Since POINTV uses NXV and NYV only once, the named error cells will still contain indications of off-scale errors when control is returned from POINTV to the calling program. (The examples assume that the scale factors have already been established.)

Example 1

```
        CALL SCERRV  (KX, KY)
        DO 300 I = 1, N
        CALL POINTV  (X, Y, NS)
        IF  (KX+ KY) 700,300,700      Test for non-zero KX and/or KY
300 CONTINUE



700 CALL DUMP                         Dump when an off-scale point is encountered.
```

Example 2

```
        .
        .
        CALL SCERRV  (KX, KY)
        M1= 0
        M2= 0
        DO 300 I = 1, N
        CALL POINTV  (X, Y, NS)
        M1= M1 + KX
300     M2= M2 + KY
```

In Example 2, M1 and M2 will contain the total number of points that are off-scale. The contents of these locations may be printed out at the end of the job or after all lines have been plotted on each frame.

SAVING AND RESETTING ERROR INDICATOR CELLS: SERSAV, SERREV

As has been pointed out, the locations assigned by SCERRV will be used as error indicator cells until new ones are named (or a new link of a chain job is loaded). As a result, subsequent executions of NXV, NYV, whether on the same level or on a lower or higher one, will alter the contents of the error cells (setting them to 1 for an unsuccessful conversion, to 0 for a successful conversion.) The only two levels involved are NXV and NYV themselves, and POINTV.

Two problems can occur. The obvious one is that indications of error may be masked by a subsequent execution of NXV, NYV. This can usually be avoided if the programmer completes error tests before there is any possibility that the contents of the error cells might be changed.

The less obvious problem is that the error cells named in one level of a program will not be accessible for error testing in other levels unless special action is taken. One way to solve this problem is to carry the error cell names in the call statements of the subprograms, not always a desirable solution.

The modules SERSAV and SERREV, used in conjunction with SCERRV, offer a convenient means for avoiding these problems in multi-level jobs. The call statements are:

> CALL SERSAV (LOCX, LOCY)
>
> This subprogram saves, in LOCX and LOCY, the locations of the cells that are currently being used for off-scale indicators.
>
> CALL SERREV (LOCX, LOCY)
>
> This subroutine resets the off-scale error cell locations that were saved by SERSAV.

The arguments LOCX and LOCY, must be variable names (either fixed or floating point) which are not used for any other purpose between the execution of SERSAV and of SERREV.

The following example shows how the programmer can use these modules to insure that errors occurring in a subprogram will not mask out errors indicated but not yet tested in a main program.

Main Program	Subprogram
.	
.	
.	
CALL SCERRV (LX1, LY1)	
Names two error indicator cells.	
.	
.	
.	
IX = NXV(X)	
IY = NYV(Y)	
Error indications will be set in LX1, LY1.	
.	
.	
.	
CALL SUBR	SUBR
	.
	.
	.
	CALL SERSAV (LOCX, LOCY)
	CALL SCERRV (LX2, LY2)
	.
	.
	.
	IX = NXV(X)
	IY = NYV(Y)
	Error indications will be set in LX2, LY2
	.
	.
	.

Main Program	Subprogram
	Test LX2, LY2 for error indications . . .
	CALL SERREV (LOCX, LOCY)
	Resets error cells LX1, LY1, using the information stored in LOCX, LOCY by SERSAV. . . .
LX1, LY1 will again be used as the error indicator cells. . . .	RETURN

E. PLOTTING AND LINE GENERATION

PLOTTING DATA

The 64 CHARACTRON characters are shown in Figure 2-32 along with the selection code for each. Note that, for FORTRAN characters, the selection codes are the same as the BCD codes recognized by IBM equipment.

Reduced to fundamentals, printing or plotting of one CHARACTRON character involves the selection and display of that character at a specified position on the raster. (A basic subprogram, PLOTV, can be used to select and display one character at a time.) The higher-level subprograms, however, contain features that make each one suitable for a specialized purpose; plotting, printing, or labeling These specialized features of the higher-level routines make them appear to be distinctly different from each other.

POINT PLOTTING SUBPROGRAMS

Since point plotting usually involves the scaled representation of a physical phenomenon, most point plotting subprograms accept physical data coordinates for position information. During execution of these subprograms, the data coordinates are converted to raster coordinates. Scale factors must have been established for the plotting routines to use in making these conversions; this requirement can be satisfied by a prior entry to GRID1V.

Scale factors are not normally saved from link to link in a chain job. Consequently plotting should be done within the link in, which scale factors are computed.

PLOTTING AN ARRAY: APLOTV

APLOTV was designed for situations in which a large number of X values are stored in one array and the corresponding Y values are stored in another array. It is possible to plot the entire set of data with one entry to APLOTV. If desired, only a portion of the data can be plotted. It is also possible to use different CHARACTRON characters as symbols to identify curves. The programmer furnishes a table (of one or more CHARACTRON characters) that will be selected cyclically.

APLOTV also keeps a tally of the number of off-scale points encountered.

The calling statement is:

CALL APLOTV (±N, XARRAY, YARRAY, JX, JY, ±NC, MARKPT, IERR)

N — Controls the number of points to be plotted. N is actually the number of points if all the data points in the arrays are to be plotted in succession. The value of N may be computed by letting K in the following formulas equal the number of points to be plotted:

N= K*JX or N= K*JY, whichever is larger.

In the FORTRAN II library version, N usually is positive, to indicate that the data arrays are in decreasing FORTRAN order of storage. However, if the arrays are in increasing order of storage, N should be negative. In the FORTRAN I version, only FORTRAN order of storage is allowed.

XARRAY YARRAY — Normally, the names of the arrays of floating point data to be plotted. Since these arguments must name the locations of the data coordinates of the first point to be plotted subscripts may be necessary. Both arrays must be in the order of storage.

S-C 4020 STANDARD SCIENTIFIC CHARACTERS

NUMERIC

CHAR	DECIMAL	OCTAL
0	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	10
9	9	11

ALPHABETIC

CHAR	DECIMAL	OCTAL
A	17	21
B	18	22
C	19	23
D	20	24
E	21	25
F	22	26
G	23	27
H	24	30
I	25	31
J	33	41
K	34	42
L	35	43
M	36	44
N	37	45
O	38	46
P	39	47
Q	40	50
R	41	51
S	50	62
T	51	63
U	52	64
V	53	65
W	54	66
X	55	67
Y	56	70
Z	57	71

OTHER FORTRAN

CHAR	DECIMAL	OCTAL
=	11	13
+	16	20
–	32	40
*	44	54
/	49	61
.	27	33
,	59	73
(	60	74
)	28	34
$	43	53
BLANK	48	60

NON-FORTRAN

CHAR	DECIMAL	OCTAL
∂	10	12
"	12	14
'	13	15
δ	14	16
α	15	17
?	31	37
β	29	35
±	30	36
π	26	32
•	42	52
γ	45	55
~	46	56
d	47	57
°	58	72
∫	61	75
Σ	62	76
□	63	77

•, ∂, ~ NOT AVAILABLE IN TYPEWRITER MODE

SUGGESTED PLOTTING CHARACTERS AND COMBINATIONS

•	42	52	+	16	20	C	19	23	0	0	0	⊠	38,55	46,67
O	38	46	X	55	67	H	24	30	Ø	38,49	46,61	⊠	63,55	77,67
□	63	77	*	44	54	U	52	64	⌀	63,49	77,61			

Figure 2-32

JX, JY — Fixed point positive integers giving the increments to be added to the subscripts of XARRAY and YARRAY as each point is plotted.

NC — The number of characters in the array MRKPT to be used as plotting symbols (usually the size of the array MRKPT). Positive NC indicates that MRKPT is a normal FORTRAN array. Negative NC shows that MRKPT is stored in increasing locations in core.

MRKPT — The array that contains the selection codes of the plotting character(s) to be used. These will be used cyclically, with the first one being used again after the NCth one has been used.

To set up this information when only one character is to be used, the argument can be one of the following:

a. An integer selection code, as "42" for the plotting dot, "38" for the alphabetic letter O, etc., see Figure 2-32.

b. A Hollerith argument for one FORTRAN character, as "1HO".

c. The name of a location containing an integer, as "MRKPT".

If more than one character is desired, the seventh argument may be one of the following:

a. An array of integers read in by the I format, or generated by a series of arithmetic statements: MRKPT(1)=38; MRKPT(2)=55, etc. Such an array must have a fixed point variable name.

b. An array read in by the A format, one character to a location. Such an array is not restricted to integer-type names; it could be called PTMRK, for example, if the contents are not used in a computation.

IERR — The name of an error location supplied by the programmer. Any point that falls outside the grid drawn by GRID1V will not be plotted. Instead a count of such points will be stored in IERR.

APLOTV plots only one character per point. If a center dot is desired, APLOTV may be repeated, using the plotting dot as the symbol. The use of APLOTV is illustrated by the Figures 2-33 and 2-34.

In Figure 2-33, it is assumed that the data are stored in two arrays: X(1), X(2), ..., X(25), and Y(1), Y(2),...., Y(25). A prior entry was made to GRID1V, using the statement:

```
CALL GRID1V (1,0.0, 11.0,0.0,180.0,0.2, 5.0, 5, 4, 10, 8, 3, 3)
```

Note that the X and Y limits cover the range of the data to be plotted. Also, since GRID1V was entered first, the necessary scale factors have been established for APLOTV. The calling statement of APLOTV for this example is the following:

```
      CALL APLOTV (25, X, Y, 1, 1, 1, 38, IERR)
or    CALL APLOTV (25, X, Y, 1, 1, 1, 1HO, IERR)
```

N=25	The array size, 25 in this case, is equivalent to the number of points to be plotted. This can only be true when all points in each array are to be plotted.
X, Y	Names of the X and Y arrays (properly dimensioned).
JX=1 JY=1	Since every X versus Y is to be plotted, the fourth and fifth arguments are set to 1.
NC=1	Only one plotting symbol is to be used; hence the sixth argument is set to 1.
MRKPT	In the two calling sequences, shown, one specifies the plotting symbol by an integer, the other uses a Hollerith argument.

Figure 7-3 shows one way that APLOTV might be used to plot a family of curves. The example assumes that the XA and YA arrays are properly dimensioned. The data is stored XA(1), XA(2),...XA(10); corresponding Y's for the three curves are stored in the array YA in the order A(1), A(2),...A(10), B(1), B(2)...B(10) C(1), C(2),... C(10). PTMRK is the name of a three-word array which was loaded as the BCD equivalents (read in by the A format) of the characters O, X, and *.

In the example, A(1), B(1) and C(1) were plotted versus X(1) each time APLOTV was entered. A DO loop was used to proceed to the next value of X, so that a total of 10 entries were made to APLOTV. The coding was:

```
          DO 1240 I=1,10
     1240 CALL APLOTV (30, XA(1), YA(1), 0, 10, 3, PTMRK, IERR)
```

Note that the plotting symbols are used cyclically, returning to the first one when the array PTMRK is exhausted. If desired, a center plotting dot can be superimposed upon the plotting symbol by repeating the entries to APLOTV with the plotting dot used for PTMRK.

PLOTTING INDIVIDUAL DATA POINTS: POINTV

For each entry to POINTV, one symbol is plotted (with or without a center dot). The coordinates may be specified as floating point data, which POINTV will convert into raster coordinates. Scale factors must have been established; this can be accomplished by a prior entry to GRID1V.

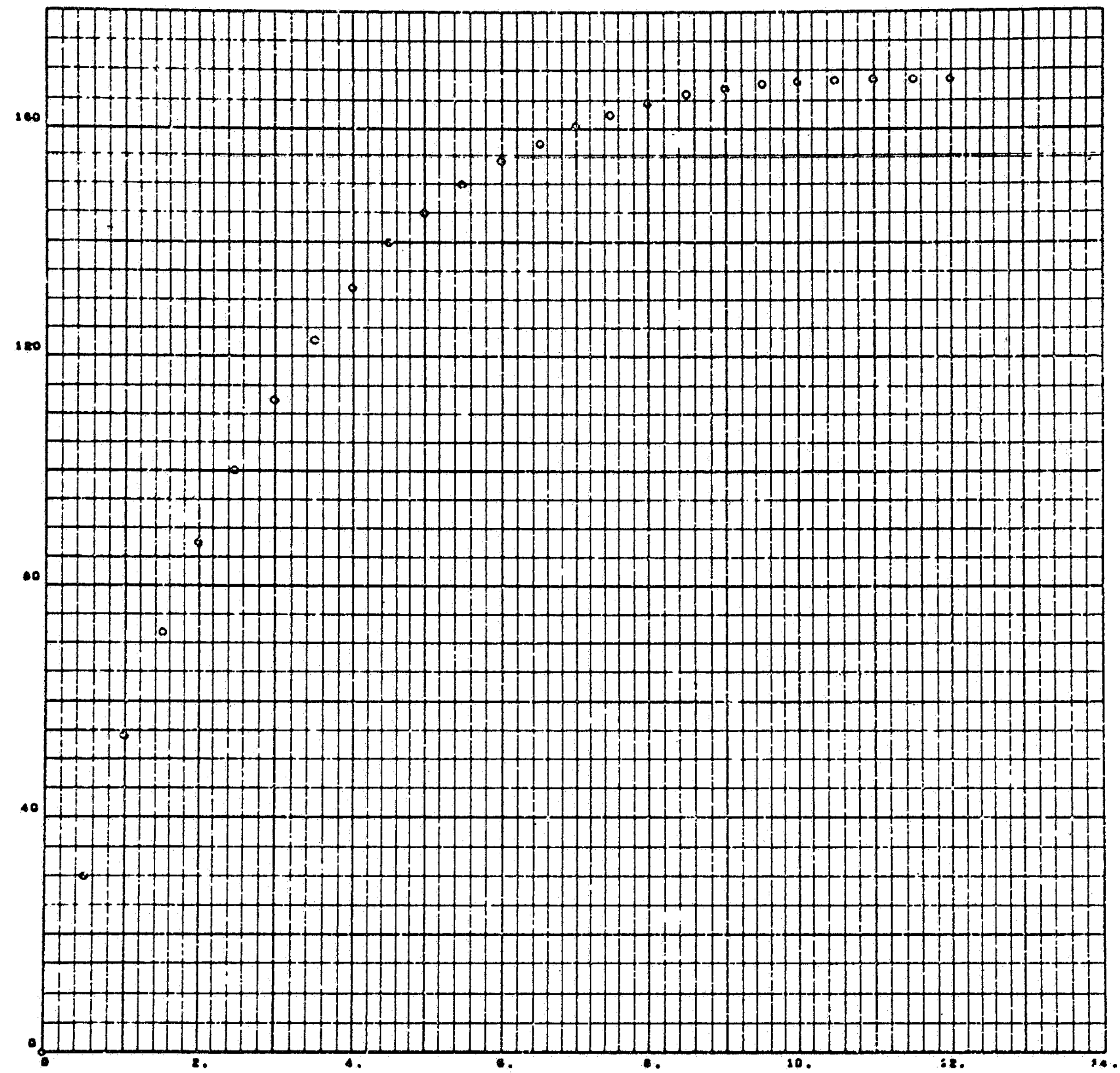
POINTV relieves the programmer of the burden of setting up a symbol table. The most suitable plotting symbols derived from CHARACTRON characters are arranged in a table, with the best choices near the beginning of the group. The symbols are selected by using integers from 1 to 48 as shown below. The calling statement is:

```
     CALL POINTV (X, Y, ±NS)
```

X, Y	Coordinates of the point to be plotted, stated as floating point data values.
±NS	Integer which selects a plotting symbol. If NS is minus, there will be no center dot plotted. If NS is positive, a center dot will appear. The value zero will be a dot on the Lockheed package only.

DATA IS STORED X1,X2,...X25 AND Y1,Y2,...Y25

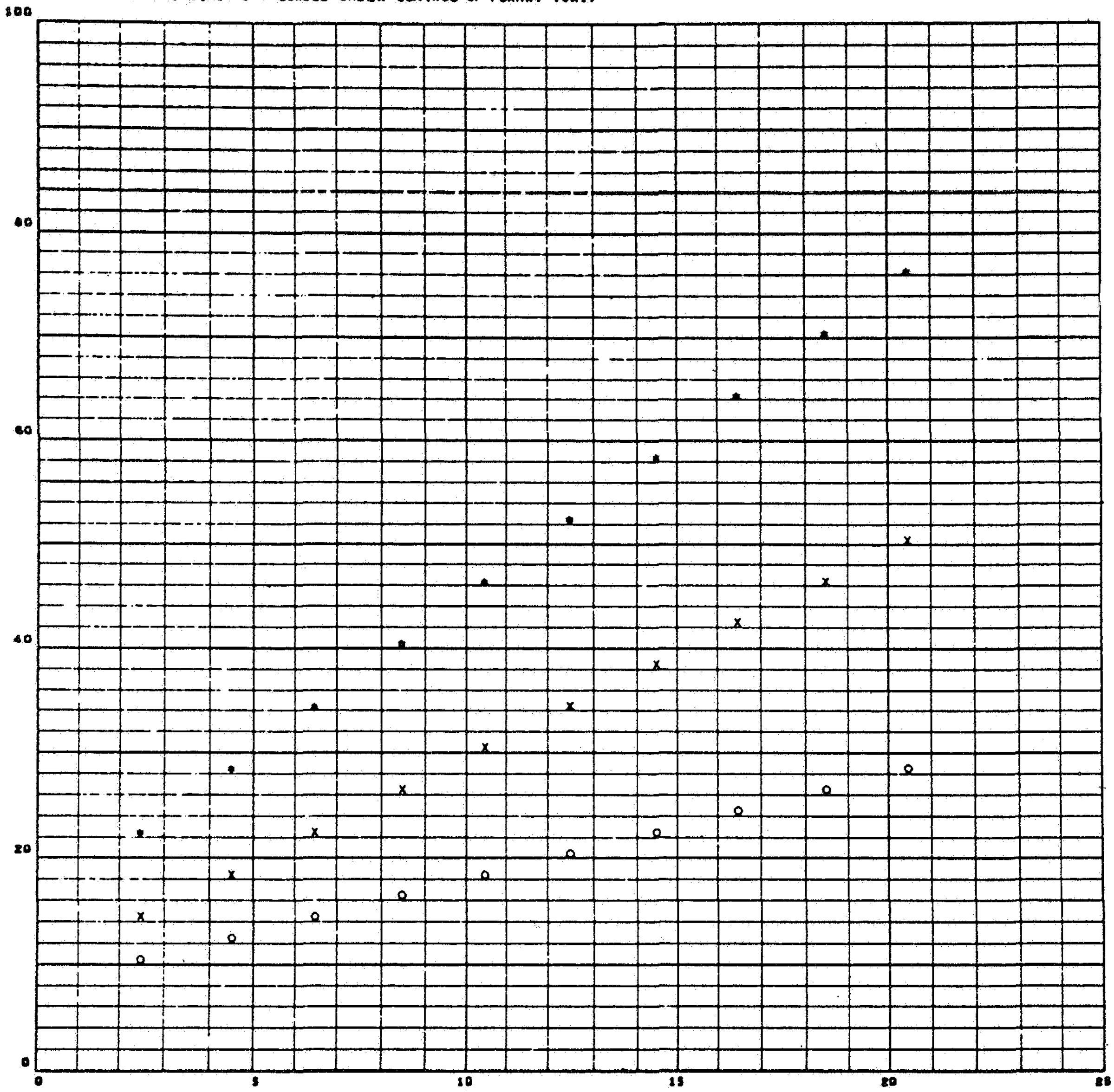
0000-00
016 016



CALL GRIDIV (1, 0.0, 14.0, 0.0, 180.0, 0.2, 5.0, 5, 4, 10, 8, 3, 3)
CALL APLOTY (25, X, Y, 1, 1, 1, 38, IERR) OR CALL APLOTY (25, X, Y, 1 1, 1, 1HO, IERR)

Figure 2-33

```
XA ARRAY IS STORED X1,X2,...X10   YA ARRAY IS STORED A1,A2,...A10 B1,B2,...B10 C1,C2,...C10        0000-00
PTMRK CONTAINS O,X,AND * LOADED UNDER CONTROL OF FORMAT (3A1)                                       020   020
```




```
     DO 1240 I=1,10
1240 CALL APLOTV ( 30, XA(I), YA(I), 0, 10, 3, PTMRK, IERR)
```

Figure 2-34

Points outside the scaled area will not be plotted. Until the programmer learns how to detect off-scale points (by a method explained in the advanced material), he should be sure that the data coordinates will fall within the limits of the scaled area.

An alternate version of this subprogram allows the programmer to specify position information in raster coordinates. This is particularly useful when the programmer wants to construct a legend in the margin, showing the symbols used and their meaning. The alternate call statement is:

CALL POINTV (IX, IY, ±NS, ANY)

IX, IY	Raster coordinates of the point to be plotted; fixed point.
±NS	Integer to select a plotting symbol, as stated above.
ANY	The inclusion of a fourth argument signals the subroutine that the position information is specified as raster coordinates. The form of this argument is immaterial; it may be any fixed point or floating point variable or constant.

Available symbols are shown in Figure 2-35. An illustration of the use of POINTV appears in Figure 2-36.

LINE GENERATION*: LINEV, LINE2V, VLAGM, VLAGX and VLAGY

LINEV connects two points by a straight line composed of vectors, joined end-to-end. The arguments for LINEV, which specify the points to be connected, must be given in raster counts. As described above, the programmer may connect two data points by a line if he first uses the functions NXV and NYV to convert the data coordinates into raster coordinates. (If there is a possibility that the

---

*Subroutines LINE2V, VLAGM, VLAGX and VLAGY have limited usage in that they are not available in all three packages.

data points being converted may be off-scale, the conversion results should be tested for errors before LINEV is executed). The calling sequence is:

CALL LINEV (IX1, IY1, IX2, IY2)

where: IX1, IY1 Raster coordinates of one end point.

IX2, IY2 Raster coordinates of the other end point.

Figure 2-37 contains an illustration of the use of LINEV.

LINE2V, which is available only in the SCORS II and IV packages, is used to draw a line from a fixed point in some direction specified by DX and DY. The calling sequence is:

CALL LINE2V (IX1, IY1, $\pm$IDX, $\pm$IDY)

where: IX1, IY1 Raster coordinates of starting point

$\pm$IDX, $\pm$IDY Number of raster points that the line is to be extended in the X and Y directions.

In either LINEV or LINE2V, a floating point data value may be utilized, if scaling has been established, by utilizing the function subprograms NXV, NYV as follows:

CALL LINEV (NXV(X1), NYV(Y1), NXV(X2), NYV(Y2))

Figure 2-38 is an example of the use of LINE2V, where each line is produced by incrementing IX1(IY1) by $\pm$ IDX ($\pm$IDY).

<u>VLAGX and VLAGY</u>, which are contained only in the Lockheed package, enable the programmer to use the variable-length axis feature to generate partial horizontal or vertical lines. (This feature generates horizontal and vertical lines that are approximately one-half the density of vector lines generated by LINEV.) These lines should not be shorter than fifteen raster counts. The call statement for sweeping a variable-length horizontal line is:

CALL VLAGX (IX, IY, LENGTH)

while the call statement for sweeping a variable length vertical line is:

CALL VLAGY (IX, IY, LENGTH)

where	IX, IY	Raster coordinates of the origin of the line.
	LENGTH	The length of the horizontal or vertical line desired from point IX, IY. LENGTH must be positive.

DASHED LINE SUBPROGRAMS: DOTLNV, DSHLNV, INCRV

The calling sequence for SCORS II and SCORS IV is:

CALL DOTLNV (IX1, IY1, IX2, IY2)

while the calling sequence for the Lockheed package is:

CALL DSHLNV (IX1, IY1, IX2, IY2)

where:	IX1, IY1	Start coordinates (integer values)
	IX2, IY2	Stop coordinates (integer values)

The routine INCRV is used to supply the line and space size. This is set normally at 8 for the line and 4 for the space. These values may be altered, however, by calling INCRV prior to DOTLNV (DSHLNV).

Calling sequence:

CALL INCRV (IL, IS)

where:	IL	Desired length of line in raster units (integer value)
	IS	Desired size of space in raster units (integer value)

If IL and IS are such that the stop coordinates (IX2, IY2) terminate in the space portion of a dash segment, the last line portion extends to the stop coordinates. Illegal values of IL or IS are ignored by INCRV. DOTLNV (DSHLNV) will then use the last used values of IL and IS.

NS	SYMBOL	NS	SYMBOL	NS	SYMBOL	NS	SYMBOL
0	•	13	∫	26	G	39	E
1	O	14	Z	27	S	40	F
2	X	15	⊠	28	W	41	M
3	□	16	⊠	29	Q	42	?
4	Y	17	⊕	30	(	43	R
5	+	18	⊞	31	~	44	ɣ
6	*	19	Ø	32	)	45	2
7	L	20	Ø	33	N	46	3
8	U	21	⌿	34	T	47	7
9	0	22	⊟	35	=	48	8
10	H	23	Ø	36	D		
11	C	24	⋇	37	A		
12	V	25	I	38	B		

Figure 2-35

BASIC PLOT-PRINT SUBPROGRAM: PLOTV

Any one of the 64 CHARACTRON characters can be displayed at a specified raster position by using PLOTV, the basic subprogram used as a lower-level module of other routines. However, the programmer may find it useful when other plotting or printing subprograms are not suitable. The call statement is:

CALL PLOTV (IX, IY, NS)

IX, IY	Fixed point raster coordinates at which the character will be displayed.
NS	An integer which selects the character. The selection code (decimal equivalent) of the desired character should be used (See Figure 2-32).

In the Lockheed package a zero setting for the rightmost six bits of NS will indicate to PLOTV that NS is a Hollerith literal.

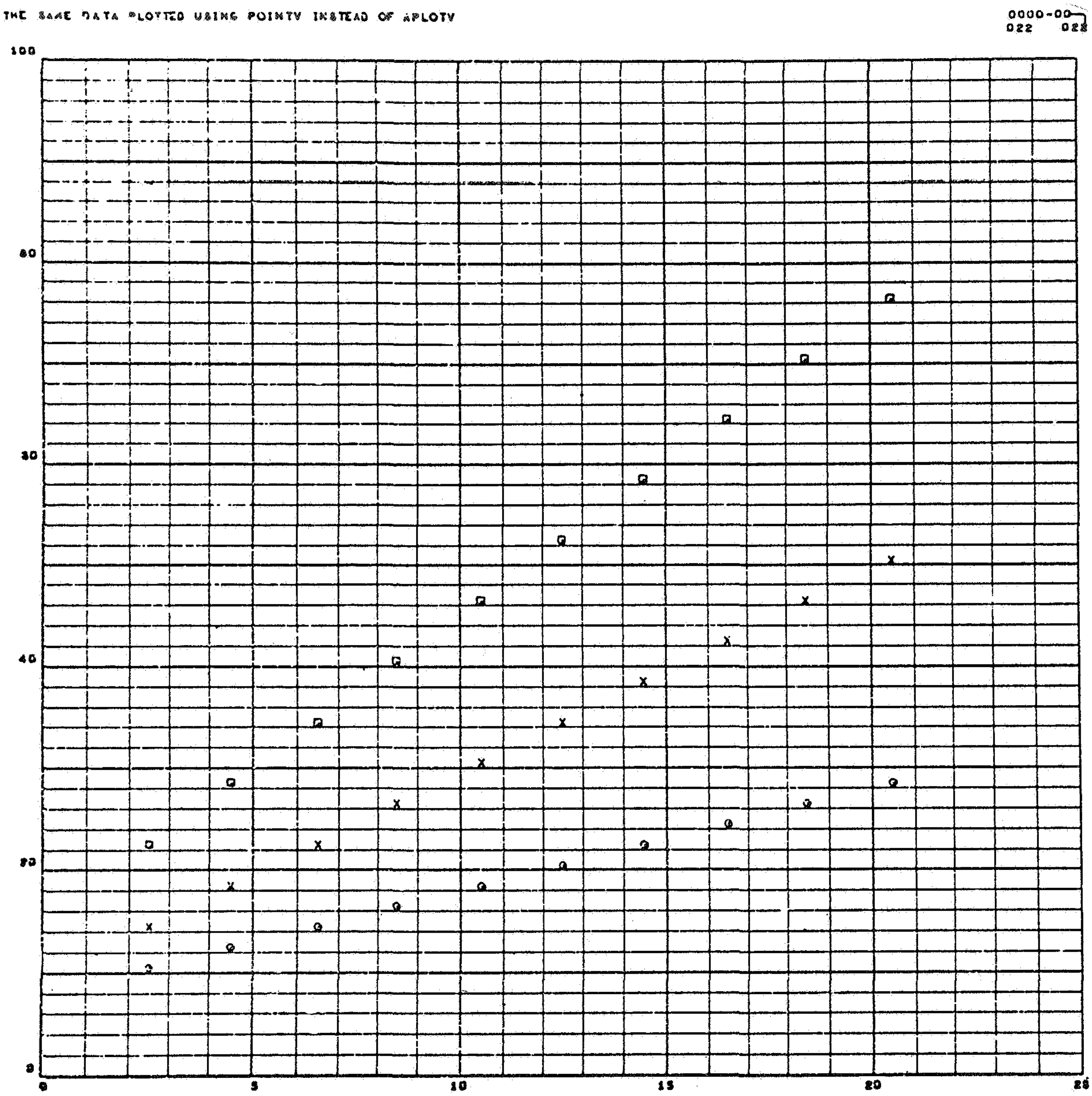
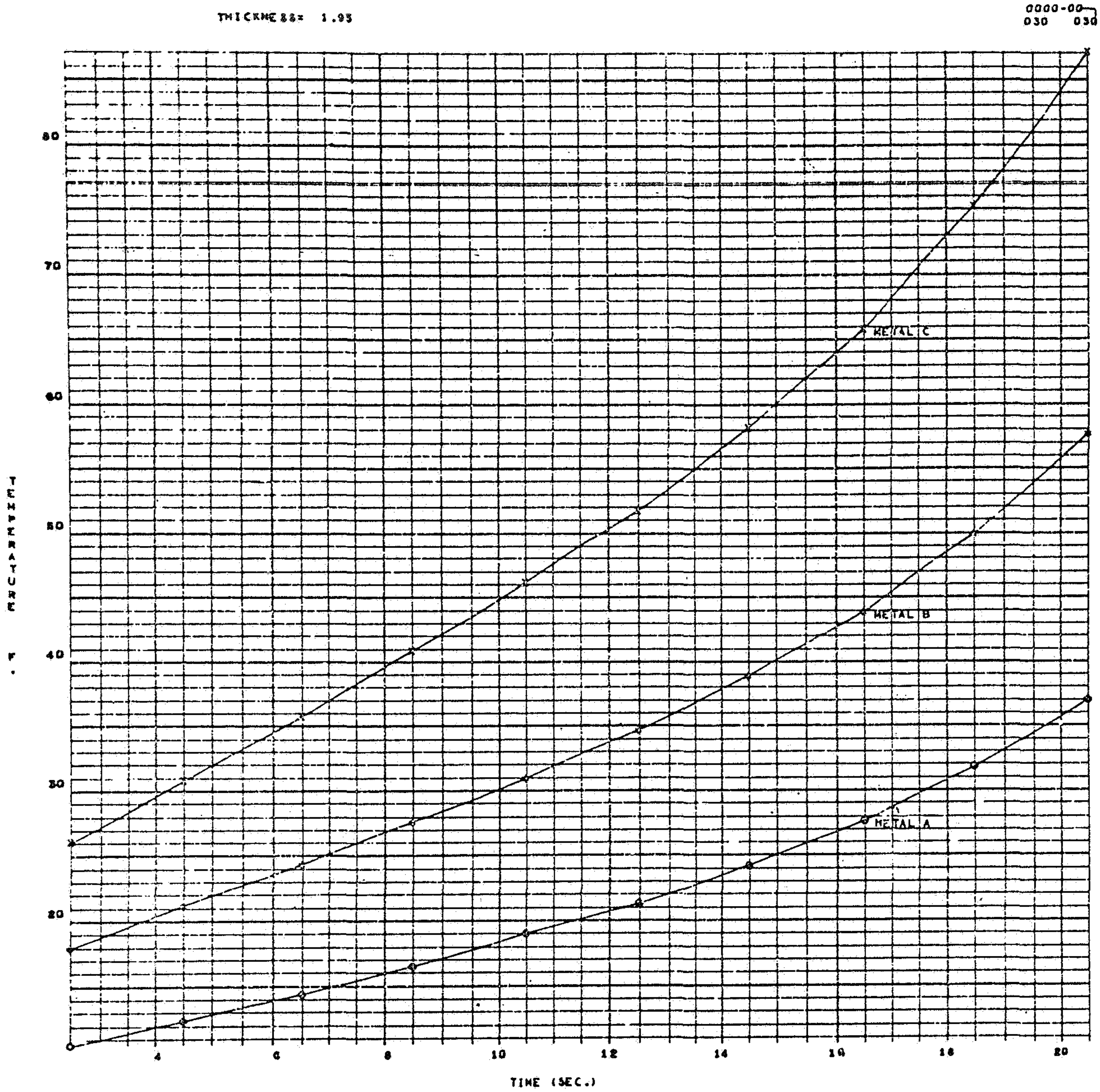

THE SAME DATA PLOTTED USING POINTV INSTEAD OF APLOTV
0000-00
022 022
100
80
60
40
20
0
0
5
10
15
20
25

Figure 2-36



THICKNESS= 1.95
0000-00
030 030
80
70
60
50
40
30
20
TEMPERATURE F.
METAL C
METAL B
METAL A
4
6
8
10
12
14
16
18
20
TIME (SEC.)

Figure 2-37

```
      CALL XSCALV (-511.0, 512.0, 0, 0)
      CALL YSCALV (-511.0, 512.0, 0, 0)
      Z0 = 0.0
      Z1 = 4.0
      Z2 = 4.0
      CALL POINTV (Z0, Z0, -16)
      DO 5 I = 1, 63
      INC = I
      JNC = -INC
      CALL LINE2V (NXV (Z1), NYV (Z0), 0, JNC)
      CALL LINE2V (NXV (Z2), NYV (Z0), 0, JNC)
      CALL LINE2V (NXV (Z0), NYV (Z1), JNC, 0)
      CALL LINE2V (NXV (Z0), NYV (Z2), INC, 0)
      Z1 = Z1 + 3.0
5     Z2 = Z2 - 3.0
```

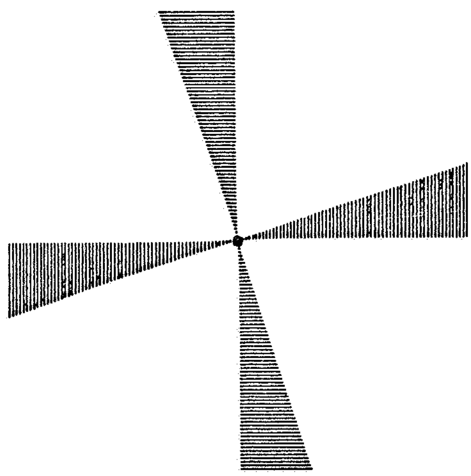


Figure 2-38

F. TITLING AND LABELING

TITLING AND LABELING SUBPROGRAMS

The printing and labeling subprograms enable the programmer to affix titles and other identifying information to a picture. Three subprograms of this type will be introduced: PRINTV, APRNTV, and LABLV. Other means for printing and labeling will be given in the section on printing.

For many applications, the positions of titles and labels must be independent of the scale. Therefore, printing and labeling subprograms accept position information in raster coordinates. This contrasts with the plotting routines, which includes facilities for the conversion of data into raster counts. Titles or labels may be positioned relative to data. The conversion functions discussed later can be employed to find raster coordinates from floating point location data.

TYPEWRITER PRINTING: PRINTV

The S-C 4020 has a built-in Typewriter Simulator, which prints a series of CHARACTRON characters very rapidly. The typewriter mode allows eight raster counts for the width of a character space. If "typewriting" exceeds 128 characters in any row and carries over to the next row, the next printing will start at the left edge of the frame, 16 raster counts below the preceding row. Writing takes place only in horizontal rows, and the characters are always upright. If the vertical advance causes the raster count to exceed 1023, printing will continue at the top of the frame, but no frame advance will occur.

In the Typewriter Mode, the machine recognizes octal codes 56 and 12 as special instructions. If these codes are accidentally used, imperfect pictures will result.

The PRINTV calling statement provides for printing characters read in by the A format, or characters specified in the call statement itself in the nH----form.

CALL PRINTV (N, BCDTXT, IX, IY)

or CALL PRINTV (-N, nH----, IX, IY)

N — The number of CHARACTRON characters to be printed. A negative sign on N signals that the second item is a Hollerith argument.

The typewriter mode is used by PRINTV until N characters have been printed.

BCDTXT — An array containing the BCD (A- or O-type format) text to be printed.

nH----- — A Hollerith argument containing the text to be printed.

IX, IY — The raster coordinates for the center of the first character. IX and IY may be any number from 0 to 1023.

VERTICAL TITLES: APRNTV

This subprogram can be used to display vertical titles composed of CHARACTRON characters. Each individual character will be upright. The call statement provides for printing characters in BCD form (read in by the A format) or characters stored as a Hollerith argument. Since APRNTV does not use the Typewriter Mode, the spacing of the characters is controlled by arguments specified by the programmer. See Figure 2-39.

The call statement for APRNTV is:

CALL APRNTV (INCRX, INCRY, N, BCDTXT, IX, IY)

or CALL APRNTV (INCRX, INCRY, -N, nH-----, IX, IY)

INCRX
INCRY — Increments used to space the characters in the X or Y direction, given in raster counts. For vertical titles, INCRX will be zero and INCRY should have a negative value. (It is suggested that INCRY fall in the range between -12 and -18 for vertical titles in most applications.) Figure 2-40 shows a vertical title, with INCRY equal to -14. Using a slightly higher value for the Y increment would have spaced letters farther apart.

The remaining arguments are as specified under PRINTV.

APRNTV was introduced and used to write vertical titles. The example in Figure 2-39 shows another application of APRNTV. This frame shows that APRNTV may be used to write along certain slopes.

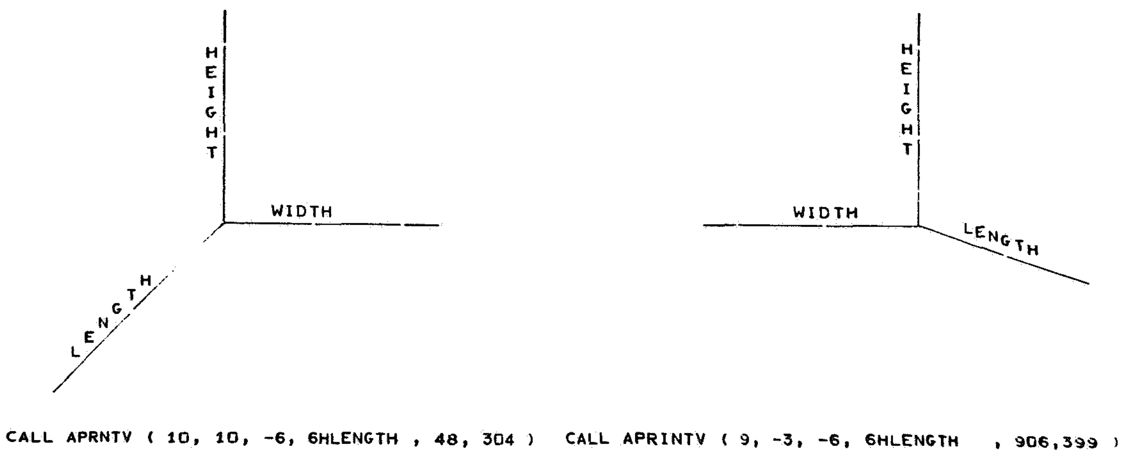


Figure 2-39

DISPLAY OF NON-FORTRAN CHARACTERS

The characters in the last column of the table of CHARACTRON characters (Figure 2-32) require special consideration. IBM equipment does not provide for key-punching these characters, so they cannot appear in a Hollerith argument nor on an A-type format card. However, the octal equivalents of the selection codes can be read into storage by the O-type format, and APRNTV or PRINTV can be employed to display them.

A single non-FORTRAN character can easily be displayed by the basic plot-print subprogram, PLOTV, or by APLOTV, using a decimal integer as the selection code. If a character is being inserted into information otherwise printed by PRINTV, the fact that PRINTV allots 8 raster counts to each character must be taken into consideration.

FIXED POINT LABELS: LABLV

LABLV was developed for GRID1V to employ in labeling grid lines, but will prove helpful when the value of computed quantities must be printed. The routine performs two chief tasks: it converts a floating point quantity into a number in BCD format, with the decimal point in the proper position, and it displays the BCD number at the raster coordinates specified.

Before the BCD label is displayed, it is right adjusted in accordance with the specified decimal scale. (The number of positions to the left of the decimal point is called the "decimal scale.") During the right adjustment, BCD blanks are moved into the high order positions that are vacated. A specified number of characters counting leading blanks and decimal point will then be displayed.

The result of this procedure is demonstrated by the column of numbers: 150. 5, 85. 7, 0. 3 in Figure 2-40. Notice that the three quantities have been truncated to the same number of decimal places, and that the decimal points have been aligned.

The calling statement of LABLV is:

CALL LABLV (D, IX, IY, NCHAR, NT, NDMAX)

D	The floating point quantity to be printed.
IX, IY	The raster coordinates which will position the first character of the label. Note that this first character may be a leading blank. If the quantity to be displayed is negative, the minus sign will be displayed one character space to the left of IX, IY.
NCHAR	Number of characters to be displayed, including leading blanks and the decimal point, if any. NCHAR is limited to 6 (or 7 if one of the characters is the decimal point).
NT	The number of times each character is to be displayed (number of over-strikes). Normally this should be 1, but 2 or more may be chosen if a darker label is desired.
NDMAX	Maximum decimal scale; i.e., maximum number of characters to be displayed to the left of the decimal point.

An integer quantity may be displayed by first changing it to floating point form and then using LABLV.

An alternate form of LABLV may be used to display labels in scientific notation. The call statement shown is used with the following changes:

-NCHAR	Number of significant figures to be displayed. NCHAR may be less than or equal to 6. The negative sign will result in the use of scientific notation.
NDMAX	May be any fixed point quantity. Since the right adjustment of these labels will not be necessary, the value of NDMAX will be ignored.

NCHAR will affect the format in the following ways:

NCHAR	FORMAT
1	Y.x10 $\pm$YY
2	Y.Yx10 $\pm$YY
3	Y.YYx10 $\pm$YY
4	Y.YYYx10 $\pm$YY
5	Y.YYYYx10 $\pm$YY
6	Y.YYYYYx10 $\pm$YY

Since the space required for these labels will be greater than that required for the fixed point format, the programmer should allow NCHAR + 7 CHARACTRON character spaces in width and 1-1/2 spaces in height as a minimum; it may be necessary to allow even more to avoid overlapping other images.

The following illustrates a scientific and a fixed point label.

CALL LABLV (579.5, 625, 1023, 5, 1, 3) RESULTS IN 579.5

CALL LABLV (579.5, 625, 999, -4, 1, 3) RESULTS IN $5.795 \times 10^{+02}$

NUMBER CONVERSION: NXV, NYV

For many applications, the placement of printing and labeling is independent of the values of the data being plotted. For this reason, the printing and labeling subprograms have been designed to accept position information specified directly in raster coordinates. In addition, many other S-C 4020 subprograms, particularly the most basic ones, are designed to accept position information specified in raster coordinates. For subprograms of this type, it is possible to position a display relative to data values by making use of the two conversion functions, NXV and NYV, described in the section on scaling.

EXAMPLES OF FRAMEV, PRINTV, APRNTV, AND LABLV

HORIZONTAL TITLES USING PRINTV

VERTICAL TITLE USING APRNTV

MACH NO.= 1.3

150.5
85.7
0.3

THE FOLLOWING FORTRAN STATEMENTS WERE USED FOR THESE EXAMPLES

```
      CALL FRAMEV
      CALL PRINTV (-31, 31HHORIZONTAL TITLES USING PRINTV  , 200, 925 )
      CALL APRNTV (0,-14,-27, 27HVERTICAL TITLE USING APRNTV  , 0, 900 )
      CALL PRINTV (-9, 9HMACH NO.=   , 200, 850 )
C     ASSUME VALUE IN XMACH IS 1.3(FLOATING POINT)
      CALL LABLV ( XMACH, 280, 850, 3, 1, 1 )
C     ASSUME D(1), D(2), AND D(3) CONTAIN 150.5, 85.7 AND 0.3, RESPECTIVELY (FLOATING POINT)
      IY = 775
      DO  410  J=1, 3
      CALL LABLV ( D(J), 200, IY, 5, 1, 3 )
  410 IY = IY-16
```

Figure 2-40

G. PRINTING

S-C 4020 PRINTING

Printing can be accomplished by the S-C 4020 in any of three ways: (1) simulated typewriting, (2) plotting single characters, (3) forming characters from vectors. The obvious application of placing titles and labels of graphs has already been introduced using the routines PRINTV, APRNTV, and LABLV. PRINTV is the basic routine for using the typewriter simulator. This is the fastest means of getting information out of the computer. In this, mode information is stored as a string of characters, six to the word, and can be output at a maximum rate of 17,000 characters/second or approximately 7,000 lines/minute. Of course, this rate is never accomplished due to the necessity for frame advances every sixty-four lines. In APRNTV, the information is output one character per word transferred to the S-C 4020. This is considerably slower than the simulated typewriter mode, but allows printing other than in straight horizontal lines.

This section of the manual presents the details of operation of the S-C 4020 typewriter simulator, introduces some additional control routines for use in printing applications, and routines forming characters from vectors. The vector method of forming characters is very flexible and provides for a variety of type fonts in selected sizes and aspect ratios as well as in four rotations. This is the only means at the programmer's disposal for producing rotated characters.

SCOUTV and VCPS

In addition to the basic routines, the routines SCOUTV and VCPS (the latter a separate routine available only in the SCORS packages, and not in the Lockheed package), will be described to aid the programmer in page formatting. The routine SCOUTV is used to output normal FORTRAN printed output on the S-C 4020. This routine utilizes the output routines implemented for the IBM 7094 or

UNIVAC 1107/1108 FORTRAN systems. The VCPS routine provides the capability of preparing printed pages with justified margins and proportional letter spacing. It also provides for mixing fonts at will to achieve special effects. These routines are described in the following two sections.

TYPEWRITER MODE

The typewriter mode, a "built-in" feature of the S-C 4020, has the following characteristics:

1. CHARACTRON characters are displayed.

2. Once positioned and started in the typewriter mode, the S-C 4020 prints character after character, automatically spacing 8 raster counts to the right as each character is printed.

3. The commands for the typewriter mode (except for those that start and stop typing) are simply the 6-character BCD words to be typed, transmitted without alteration. This mode, therefore, is very economical of computer time and the length of the tape used for words in the typewriter mode is held to a minimum. Since the S-C 4020 can display 6 characters per command, this mode is also economical with S-C 4020 time.

4. When a line of typing reaches the right edge of the frame, (128 characters), an automatic carriage return takes place. Typing halts at this point and resumes at the left edge of the frame, 16 raster counts below the preceding line.

5. If typing reaches the bottom of the frame, a return to the top of the frame occurs automatically. No automatic film advance takes place, so it is possible for typing to overlay previously displayed information.

6. Three of the 64 CHARACTRON characters serve special purposes when they are encountered in the words of the text being typed.

41 (Decimal) 52 (Octal)	A carriage return, described in item 4, will occur when this character is encountered.

10 (Decimal)
12 (Octal) — "Stop type". Typing mode is stopped, any characters remaining in the word will be ignored. The next word on tape will be interpreted by the S-C 4020 in the graphic mode.

46 (Decimal)
56 (Octal) — "Reset" is executed. This means that typing is stopped, the film is advanced, and the bright intensity mode is turned on. Any characters remaining in the word will be ignored. The next word on tape will be interpreted in the graphic mode.

Because 42, 10, and 46 are used for these special purposes, the characters · (plotting dot), ∂ , and ~ are unavailable in the typewriter mode.

7. Typing may be started at either a specified point or at the "current point". The "current point" is defined as the point specified by the last operation performed; i.e., either the last point plotted or typed, or the origin of the last vector or axis drawn.

SYSTEM SUBPROGRAMS AND THE TYPEWRITER MODE

The subprograms PRINTV and LABLV, and the SCOUTV system, make use of the typewriter mode. Therefore, the output of these subprograms will be influenced by the characteristics of this mode.

Since it was not practical to require the programmer to insert a "stop type" code other methods of stopping typing are used. PRINTV and LABLV insert the "stop type" code after the specified N characters. The SCOUTV system inserts the code in accordance with the FORMAT statement used.

It should be apparent that an incompatibility would occur if a "stop type" or "reset" code were encountered within the text being typed by PRINTV, LABLV, or SCOUTV. Any information following the special code would not be interpreted as typewriter output.

If PRINTV, LABLV, or SCOUTV printing encounters a "carriage return" character or the right edge of the frame, a carriage return will be made. This would interfere with the normal line advance provided in the SCOUTV system. It can be used to advantage, however, with PRINTV.

Except for the method used to stop typing, PRINTV can be considered a basic typewriter mode routine. It is a suitable lower-level module for other CHARACTRON character typing routines; in fact, it is used in this way by LABLV.

"STOP TYPE" ROUTINES: STOPTV, RESETV

It is remotely possible that some unusual use of the subroutines or of the S-C 4020 could cause the typewriter mode to be in effect when the graphic mode is needed. Two subroutines are provided which ensure that the typewriter mode is no longer on.

> CALL STOPTV
>
> Turns off the typewriter mode. This has no effect on the mode if the typewriter mode is already off. However, since this statement writes a word on tape, it should be used with discretion.
>
> GRID1V uses STOPTV to ensure that the plotting mode is in effect at the start of a new grid display.
>
> CALL RESETV or CALL RESETV (NF)
>
> Turns off the typewriter mode, advances the film and establishes the bright intensity mode. The argument agrees with FRAMEV.

SETTING THE "CURRENT POINT": PRINTV, BRITEV, FAINTV

The current point feature of the typewriter mode can be utilized if a special form of PRINTV is employed. Typing may be started at the last point plotted or printed or at the origin of the last vector or axis line drawn, by giving one of the following statements:

CALL PRINTV (N, BCDTXT)

CALL PRINTV (-N, nH------)

Except for the use of the current point instead of specified coordinates, this alternate form of PRINTV functions in the usual manner.

If the programmer wants to set a new current point, he may plot or print a blank at a specified position. Another method, which includes both setting the current point and establishing the "Brite" or "Faint" mode, is by one of the following statements:

CALL BRITEV (NX, NY)

CALL FAINTV (NX, NY)

NX, NY — The raster coordinates which will be set as the current point. Only a blank will be plotted at NX, NY by either of these statements.

USE OF THE FORM SLIDE: FORMV

In using the S-C 4020 for printing applications, it is desirable to provide the equivalent of using preprinted forms. For this purpose, the S-C 4020 is equipped with a FORM FLASH unit to expose the film with a prepared slide which can be precisely positioned. The CHARACTRON tube can be aligned to make the image fit the form to be used. This is a manual operation and FORM slides cannot be changed under program control.

The preparation of good FORM slides is rather expensive and therefore should only be considered for well established forms to be used hundreds of times. The FORM slide can occasionally be used for short runs if a special effect is required which is difficult or impossible to achieve by programming the S-C 4020. It must be noted here that a FORM slide should not be considered for use as a grid background for plotting since slight distortions in the image are not harmful if they affect the data and the grid at the same time but would ruin the precision of a graph drawn against a projected grid. Illustrations of the various form slides which are available at GSFC are contained in Appendix E.

The FORM is projected by the statement:

CALL FORMV

SPECIAL PRINTING AND LABELING AIDS: HOLLV, BNBCDV

HOLLV and BNBCDV were designed as lower-level modules for the system printing and labeling subprograms. However, they can be called directly, and can sometimes be useful building blocks for the programmer's character printing and labeling subprograms.

ISOLATING ONE CHARACTER FROM A BCD ARRAY: HOLLV

HOLLV selects the Nth BCD character from BCD words that are either contained in a one-dimensional FORTRAN array or given as a Hollerith argument, and stores the numerical equivalent of the selected character in a specified location. PLOTV or VCHARV can then be used to display the character. The call statement is one of the following:

CALL HOLLV (N, BCDTXT, NS)

CALL HOLLV (-N, nH------, NS)

±N	The position of the desired character within the block of BCD characters, counting from 1. If the block of characters is a FORTRAN array, N should be positive. If the BCD characters are in a Hollerith argument, the sign of N should be negative.
BCDTXT	A one-dimensional FORTRAN array that contains the BCD characters.
nH---	A Hollerith argument

NS The name of a location in which the numerical equivalent of the selected character will be stored. For example, if the Nth BCD character is R, the stored result will be the same as if the statement "NS = 41" had been used.

A practical example of HOLLV usage is illustrated by the way APRNTV uses it. The latter subprogram loops through a sequence of BCD characters, using HOLLV to select a character and PLOTV to print it. APRNTV then advances the print position and returns to the beginning of the loop for the next character.

OBTAINING BCD EQUIVALENT OF A FLOATING POINT NUMBER: BNBCDV

BNBCDV is used by LABLV as a lower-level module to obtain the BCD equivalent of the floating point number displayed in the label. Since it converts a floating point number so as to yield the six most significant digits and the decimal scale, it can be useful also in programs that do not involve the S-C 4020. The call statement is:

CALL BNBCDV (D, BCDWD, NDS)

D Floating point number to be converted into BCD.

BCDWD Location in which the 6-character (rounded) BCD equivalent will be stored.

NDS Location in which the decimal scale (number of digits to the left of the decimal point) will be stored.

The sign of D is ignored, and must be handled separately by the programmer.

Examples

D = 12.345678

The BCD equivalent of 123457 will be placed in BCDWD, and the decimal scale of 2 will be stored in NDS.

D = -.01234567

The BCD equivalent of 123457 will be placed in BCDWD, and the decimal scale of -1 in NDS.

PROGRAM-GENERATED CHARACTERS: RITE2V, VCHARV, CHSIZV, RITSTV

VECTOR CHARACTER WRITING SUBROUTINE: RITE2V

The RITE2V subprogram provides a means for printing variable-size characters in one of four different orientations. It uses, on a lower level, the vector character-generating subprogram, VCHARV.

The call statement can be either of the following:

CALL RITE2V (IX, IY, LIMIT, K, INT, NTOTAL, NTH, BCDTXT, NLAST)

CALL RITE2V (IX, IY, LIMIT, K, INT, NTOTAL, -NTH, nH-----, NLAST)

IX, IY	The raster coordinates used for positioning the center of the first character to be printed.	
LIMIT	A raster position which will determine the right margin. When this limiting position is reached, writing is halted and a new line is started below the starting position of the previous line.	
K	A code number which selects the direction of writing and the orientation of characters within the written text.	
K=90	Characters will be upright, as	A
	Writing will be horizontal, left to right, as	AND
K=180	Characters will be rotated left, as	A
	Writing will be vertical, bottom to top, as	AND

K=270	Characters will be upside down, as	∀
	Writing will be horizontal, right to left, as	ᗡNⱯ
K=0 (or 360)	Characters will be rotated right, as	A
	Writing will be vertical, top to bottom, as	AND

INT An integer which controls intensity of the characters by controlling the number of times each vector is displayed. If "overstriking" for a darker character is not required, INT should be 1.

NTOTAL An integer which specifies the number of characters to be printed.

±NTH A character position count that specifies the location within a BCDTXT where writing is to begin. (Usually this is 1, but this argument allows printing to begin with some character in the BCDTXT other than the first.)

The sign indicates the type of storage used for the next argument. NTH should be positive if the 8th argument is a FORTRAN array. A negative value should be used if it is a Hollerith argument.

BCDTXT The name of an array of BCD information to be written.

nH--- A Hollerith argument. If used, a negative sign must be used with NTH to signal the subroutine that the information is in this form.

NLAST The name of an error location (fixed point). NLAST will be set to zero if all the NTOTAL characters have been displayed. If writing must be stopped because the end of the frame has been reached, NLAST will be set to the character count where writing was stopped.

The standard character size is approximately 12 raster counts in width by 18 counts in height. The standard spacing allowance for adjacent characters is

18 raster counts, and for adjacent rows is 26 raster counts.

The programmer can alter the size of characters and spacing by means of other subroutines, discussed later in this section. He will not need to use them if the standard specifications are satisfactory for his purposes. Figure 2-41 illustrates the use of RITE2V to title a graph on any margin.

VARIATION IN SIZE OF VECTOR CHARACTERS: CHSIZV

The width and height of vector characters can be altered by the subroutine CHSIZV ("Change Size V"). The call statement is:

CALL CHSIZV (LVW, LVH)

LVW	An integer, from 1 to 15, that controls the width of characters.
LVH	An integer, from 1 to 9, that controls the height of characters.

Obviously, LVW and LVH need not be equal, but any great disparity will result in the formation of characters that are elongated in one direction.

For the usual applications of RITE2V, the character size can be considered to be 4*LVW in width, and 6*LVH in height. These dimensions do not include the space required to prevent overlapping of adjacent characters. Examples of different proportions are shown in Figure 2-42. Notice that minimum size characters are not easily read. Adjustment of vector intensity is very critical.

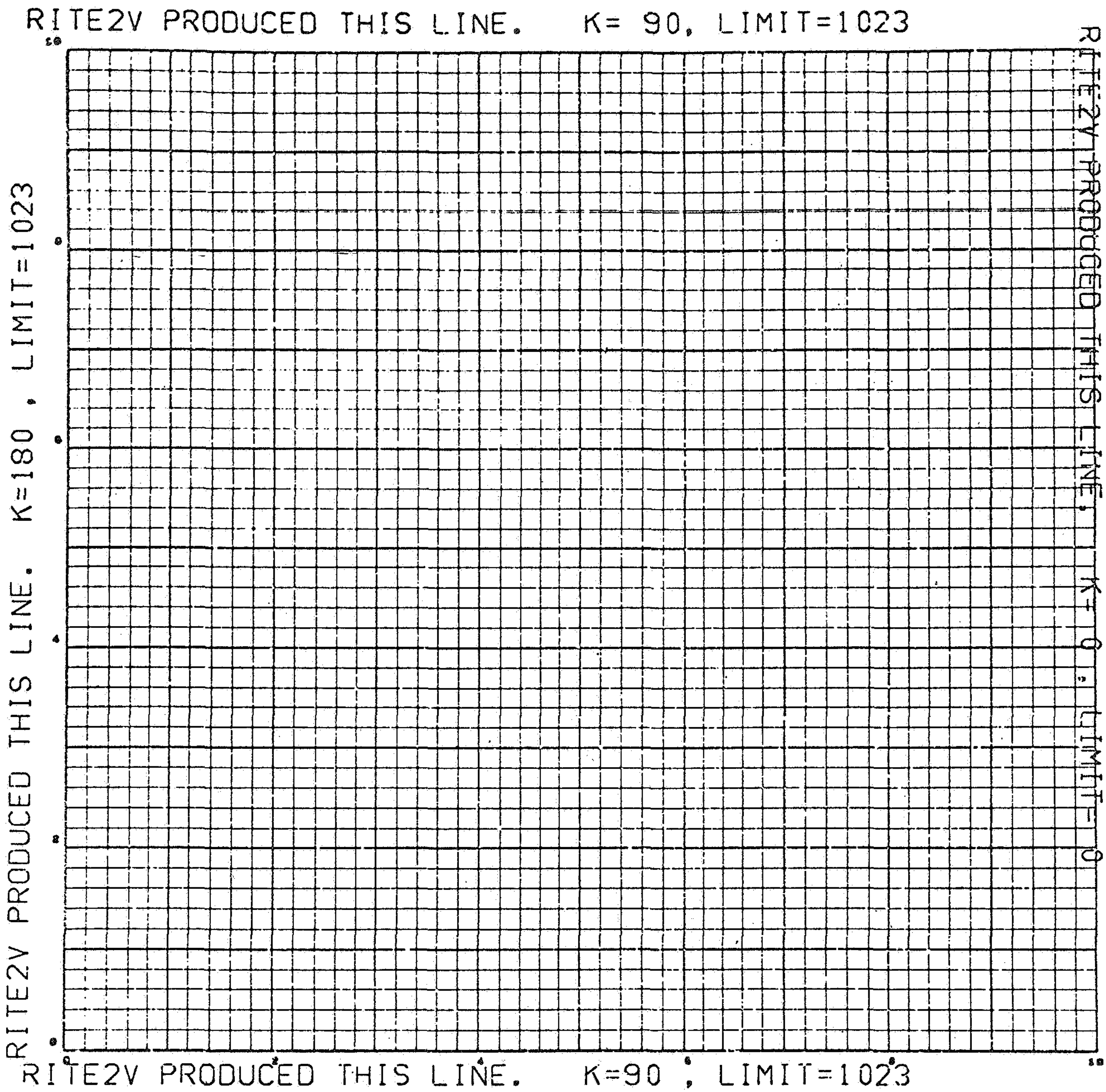

RITE2V PRODUCED THIS LINE. K= 90, LIMIT=1023
RITE2V PRODUCED THIS LINE. K=180 , LIMIT=1023
RITE2V PRODUCED THIS LINE, K= 0 , LIMIT= 0
RITE2V PRODUCED THIS LINE. K=90 , LIMIT=1023

Figure 2-41

If CHSIZV is never called, the standard values of LVW=3 and LVH=3 will be used, resulting in a character size of 12 x 18 raster counts.

When CHSIZV is used to alter the standard character size, it will probably also be necessary to adjust the character spacing. The subroutine RITSTV was designed for this purpose.

CHARACTER SELECTION AND SPACING: RITSTV

RITSTV can be used to alter spacing used by RITE2V for characters and rows. In addition, RITSTV can be used to select an alternate character set to be displayed by RITE2V. The call statement is:

CALL RITSTV (ISPACE, IROW, TABL_V)*

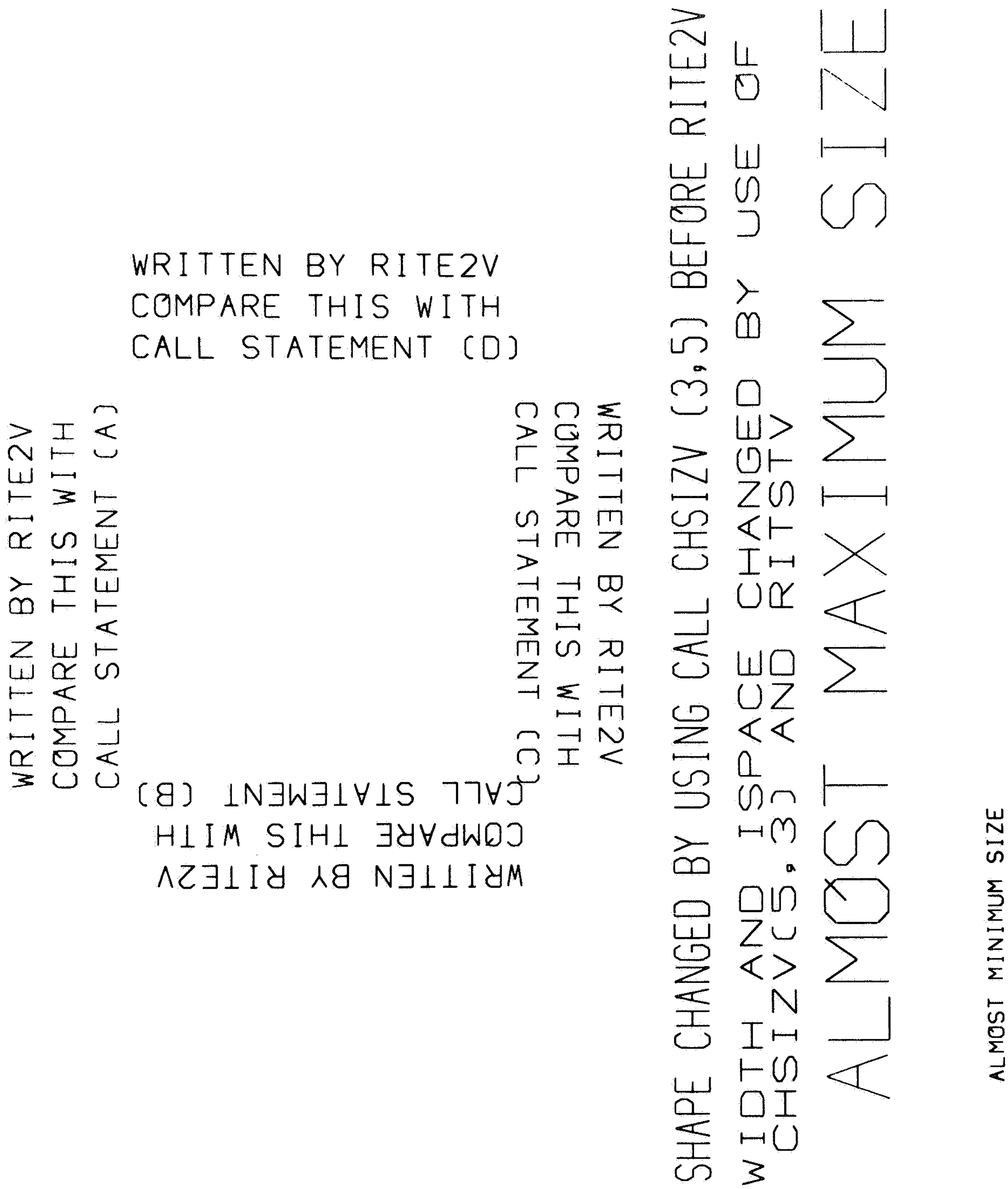
ISPACE — An integer that specifies the number of raster counts allowed for adjacent characters. This should be somewhat larger than the width of each character, to allow a gap between them.

IROW — An integer that specifies, in raster counts, the depth of adjacent rows. This should be somewhat larger than the height of a character.

TABL_V* — The name of a table of character patterns to be used. The most frequently used table, containing the alphanumeric characters, is named TABL1V. The available tables are described in the section entitled Available Vector Character Tables.

WHEN RITSTV IS USED, THE TABLE NAME GIVEN AS THE THIRD ARGUMENT MUST ALSO BE GIVEN ON AN "F" CARD IN FORTRAN II or "EXTERNAL" IN FORTRAN I

NOTE: TABLE names larger than 9V are designated TAB--V instead of TABL V. References to TAB--V in this section of the manual are marked with an asterisk as a reminder of this fact.



(Please turn page sideways).

Figure 2-42

If RITSTV is called, all three arguments must be specifically stated, even though not all of them are changed, and the table name must appear on an "F" card or an "EXTERNAL" definition. The vector character tables are illustrated in Figures 2-43 and 2-44.

If RITSTV is never called, RITE2V will use the standard values of ISPACE and IROW (18 and 26, respectively), and the standard table (TABL1V).

The following formulas are a general guide for the assignment of minimum values of ISPACE and IROW.

> ISPACE $\approx$5* LVW + 3
>
> IROW$\approx$ 7*LVH + 5

RETRIEVAL OF TERMINAL COORDINATES: RITXYV

At the time RITE2V terminates, the coordinates for the next character are set up. RITXYV retrieves these coordinates so that they can be used to determine the correct position of a special character to be inserted into a text written by RITE2V. The call statement is:

```
CALL RITXYV (LX, LY)
```

LX, LY	Names of the locations in which RITXYV will store the raster coordinates (IX, IY) that RITE2V has computed as the location of the next character.

DISPLAY AN INDIVIDUAL VECTOR CHARACTER: VCHARV

VCHARV is used as a lower-level subroutine by RITE2V to generate each vector character. When a single character is to be displayed, a direct call to VCHARV by the programmer will often be more convenient than a call to RITE2V, especially if the character to be displayed is a non-Hollerith or a special character.

The call statement is:

CALL VCHARV (K, INT, ILX, ILY, NS, TABL_V*)

K	Code number which selects the orientation of the character:	
K=90	Character will be upright, as	A
K=180	Character will be rotated left, as	
K=270	Character will be upside down, as	∀
K=0 (or 360)	Character will be rotated right, as	

INT — An integer which controls the intensity of the character by controlling the number of times each vector is repeated. For normal intensity this should be 1, but may be greater if a darker character is desired.

ILX, ILY — Integers which specify the raster coordinates to be used to position the character. This position will be used as the lower left corner of the character pattern. (Note that this is different from the center position specified for RITE2V).

NS — An integer which selects the character to be generated from a table of patterns. The actual character produced will depend upon the table being used; the available tables are discussed in the next section entitled Available Vector Character Tables.

(If a Hollerith character is desired, and if TABL1V is used for the last argument, a properly stated Hollerith argument can be employed for NS. For example, the argument 3H001 is equivalent to an integer 1, 3H00A is equivalent to the integer 17 that will select A, etc.)

TABL_V* — The name of a table of character patterns to be used. The name of the table containing the most frequently used characters is TABL1V.

WHEN VCHARV IS CALLED DIRECTLY, THE TABLE NAME GIVEN MUST ALSO APPEAR ON AN "F" CARD OR ON "EXTERNAL" DEFINITION.

RITE2V uses the specified coordinates as the center of the first character to be displayed, but VCHARV uses the specified coordinates as the lower left corner of the character. Thus, it may be necessary to translate the coordinates when direct calls to VCHARV are used to insert characters into a line of information displayed by RITE2V. Without going into great detail on the structure of vector characters at this point, the formulas given below show the relation between the position arguments for VCHARV and for RITE2V. In these formulas, LVW and LVH are the arguments used in CHSIZV. If CHSIZV has not been used to alter these quantities, LVW = LVH = 3.

FOR K = 90	ILX = IX - 2*LVW
	ILY = IY - 3*LVH
K=180	ILX = IX + 3*LVH
	ILY = IY - 2*LVW
K=270	ILX = IX + 2*LVW
	ILY = IY + 3*LVH
K=0	ILX = IX - 3*LVH
(or 360)	ILY = IY + 2*LVW

AVAILABLE VECTOR CHARACTER TABLES: TABL1V, TABL2V, TABL3V, TABL5V, TABL6V, TABL7V, TABL8V, TABL9V, TAB11V, and TAB15V

NOTE: Only TABL1V, TABL2V, and TABL3V are available in the Lockheed Package. All the tables are available for the SCORS Packages.

Of the three tables of character patterns available on all system libraries, TABL1V is an alphanumeric table, TABL2V contains lower case Greek characters, and TABL3V contains the upper case Greek characters.

The first 64 characters in TABL1V are arranged in an order compatible with the Charactron character table. Any code which will select one of the Charactron characters will select the equivalent character from TABL1V. However, TABL1V is extended to include some special characters, selected by codes numerically greater than those permissible for the Charactron characters. Since codes greater than 63 cannot be expressed as BCD (or Hollerith) characters, RITE2V cannot be used to display them. However, VCHARV can be employed for any special characters that have selection codes greater than 63.

Usually VCHARV will be utilized whenever a character from TABL2V or TABL3V is to be displayed. However, it is of interest to note that the position of 0, 1, 2, ... etc., in TABL1V are equivalent to the positions of α, β, γ, ... etc., in TABL2V. Therefore, if the Hollerith argument 3H012 were to be used in RITE2V, and if TABL2V had been specified by RITSTV, the characters α, β, γ, would be displayed. TABL8V and TABL9V provide upper and lower case Greek alphabets for use with RITE2V.

REMEMBER, WHEN ONE OF THE TABL_V NAMES IS USED BY THE PROGRAMMER AS AN ARGUMENT, IT MUST ALSO BE NAMED ON AN "F" CARD OR AN "EXTERNAL" DEFINITION.

Figures 2-43 and 2-44 show the Vector Character Tables. A portion of the coding for these illustrations is shown because it illustrates the use of the tables and of VCHARV, as well as provides an example of the use of APRNTV in a special situation.

Note the use of APRNTV to produce the headings for the rows and columns. Utilization of APRNTV for this application permits the required special spacing of Charactron characters.

TABL1V,TABL2V,TABL3V --NOTE USE OF VCHARV TO DISPLAY THE CHARACTERS

TABL1V

	0	1	2	3	4	5	6	7	8	9
00	0	1	2	3	4	5	6	7	8	9
10	∂	=	"	'	δ	α	+	A	B	C
20	D	E	F	G	H	I	π	.	]	β
30	±	?	–	J	K	L	M	N	O	P
40	Q	R	•	$	*	γ	~	d		/
50	S	T	U	V	W	X	Y	Z	°	,
60	(	∫	Σ	□	&	%	¢	[	]	{
70	}	>	<	≥	≤	∞				

TABL2V

	0	1	2	3	4	5	6	7	8	9
00	α	β	γ	δ	ε	ζ	η	θ	ι	κ
10	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ
20	φ	χ	ψ	ω						

TABL3V

	0	1	2	3	4	5	6	7	8	9
00	A	B	Γ	Δ	E	Z	H	Θ	I	K
10	Λ	M	N	Ξ	O	Π	P	Σ	T	Υ
20	Φ	X	Ψ	Ω						

TABL5V

	0	1	2	3	4	5	6	7	8	9
00	!	á	é	í	ó	ú	•	?	¿	ñ
10		=	'				&	a	b	c
20	d	e	f	ø	h	i		.	]	
30		?	–	j	k	l	m	n	o	p
40	q	r		†	*					/
50	s	t	u	v	w	x	y	z		,
60	(									

TABL7V

	0	1	2	3	4	5	6	7	8	9
00	œ	!	:	¿	ñ	´	`	"	ç	Ç
10		=	"				+	a	b	c
20	d	e	f	ø	h	i		.	]	
30		?	–	j	k	l	m	n	o	p
40	q	r		?	*					/
50	s	t	u	v	w	x	y	z		,
60	(									

Figure 2-43

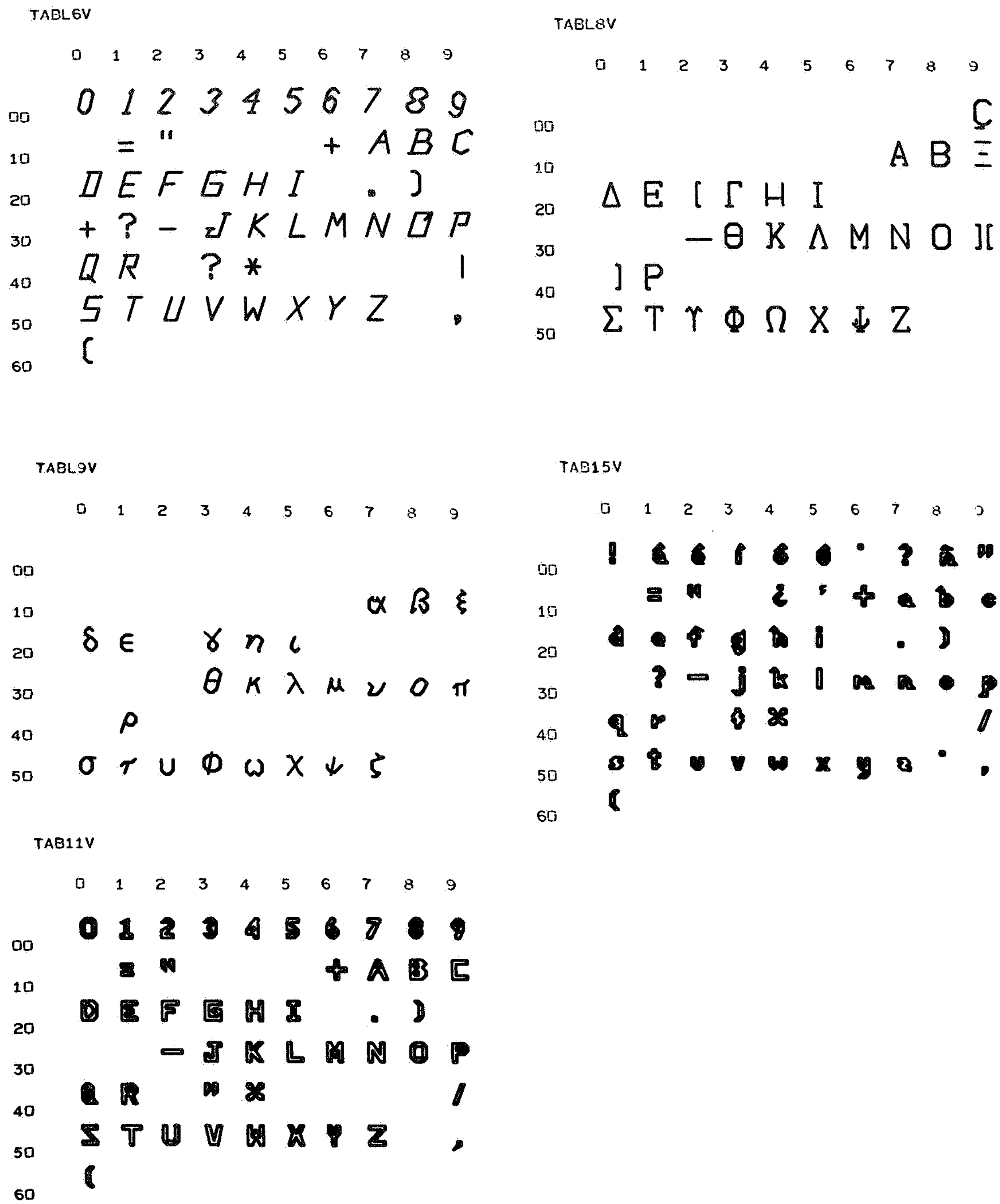


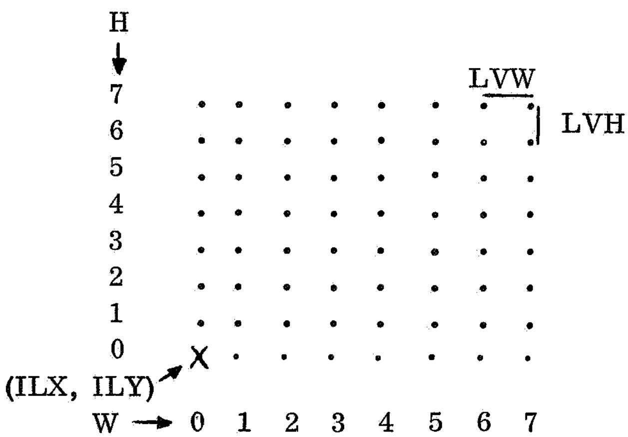
Figure 2-44

VECTOR CHARACTER CONSTRUCTION

In a preceding section, the tables of vector characters which are available in the system were presented. This section explains how vector characters are generated and how the programmer can create additional characters to be displayed by VCHARV and RITE2V.

VECTOR CHARACTER FORMATION

Given the proper specifications, VCHARV will generate any design that can be constructed by joining points on an 8 x 8 lattice with straight lines. A coordinate system in width (W) and height (H) is used to refer to positions on the lattice. Patterns are specified by giving coordinates of the positions on the lattice that are to be connected by vectors. As shown on the diagram, the W and H coordinates each can have values from 0 to 7.



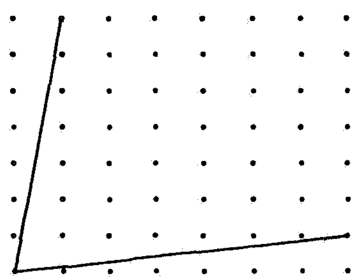
ILX, ILY are the coordinates used by VCHARV to position the character. LVW, LVH represent the distances (expressed in raster counts) from one lattice position to an adjacent position.

CHARACTER SIZE

The size of a character (or other design) depends upon the amount of space between adjacent positions on the lattice. The number of raster counts between positions may be specified by calling the subprogram CHSIZV. The arguments LVW and LVH indicate the desired number of raster counts between positions in width and height, respectively. If CHSIZV is never called, the values LVW = LVH = 3 raster counts will be used.

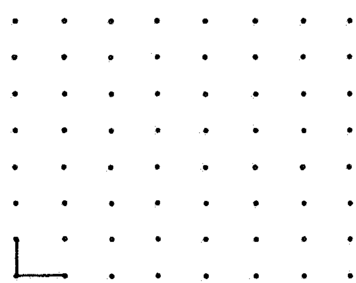
Although it is possible to assign a value as small as 1 to LVW and/or LVH, this is not practical for the majority of characters. The upper limit depends upon the particular character, and upon the restriction that no vectors in the pattern may exceed the resultant of 63 counts in the direction W, and 63 counts in the direction H. For example, if a particular pattern requires joining a position on the bottom row of the lattice with one on the top row, the maximum permissible value of LVH is 9, since 9 (raster counts) multiplied by 7 (lattice intervals) = 63 counts.

To draw there two vectors:



The maximum permissible LVW, LVH = 9 counts.

To draw these two vectors:



The maximum permissible LVW, LVH = 63 counts.

CHARACTER ORIENTATION AND PLACEMENT: VCHARV

VCHARV permits a selection of four possible orientations. It rotates a character into the selected orientation by the way in which it applies the increments LVW and LVH.

The following table shows the direction of application of LVW and LVH for the four orientations.

Orientation Selection	Directions in Which Increments are Applied	
K	LVW	LVH
90 (Upright)	Positive in X	Positive in Y
180	Positive in Y	Negative in X
270	Negative in X	Negative in Y
0 (or 360)	Negative in Y	Positive in X

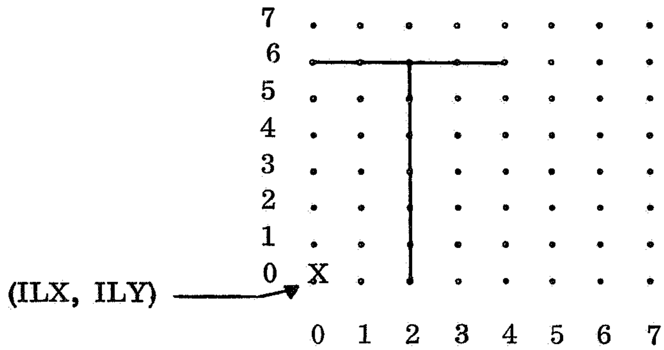
To prevent characters from overlapping the edge of the frame, the orientation and the specifications of the position coordinates (ILX, ILY) must be considered. The following table shows the restrictions upon ILX, ILY for each of the four orientations. In the table, N is one less than the number of lattice positions in width used by the character, and M is one less than the number, in height.

K	ILX	ILY
90 (Upright)	$0 \leq ILX \leq 1023 - N*LVW$	$0 \leq ILY \leq 1023 - M*LVH$
180	$M*LVH \leq ILX \leq 1023$	$0 \leq ILY \leq 1023 - N*LVW$
270	$N*LVW \leq ILX \leq 1023$	$M*LVH \leq ILY \leq 1023$
0 (or 360)	$0 \leq ILX \leq 1023 - M*LVH$	$N*LVW \leq ILY \leq 1023$

PATTERN SPECIFICATION

A pattern specifies the points on the lattice to be joined by vectors. The pattern consists of a sequence of lattice coordinates arranged in the order W1, W2, H1, H2, W3, W4, H3, H4, ..., etc. where (W1, H1) and (W2, H2) are the coordinates of the end points of the first vector in the character, (W3, H3), and (W4, H4) are coordinates of the end points of the second vector, etc. The end of a pattern is signaled by the presence of at least 12 consecutive zero bits.

Since values of W and H will not be larger than 7, the values can be packed into computer words, allotting 3 bits per value. As many words as are needed to include all coordinates may be used. The character T appears on the lattice as follows:

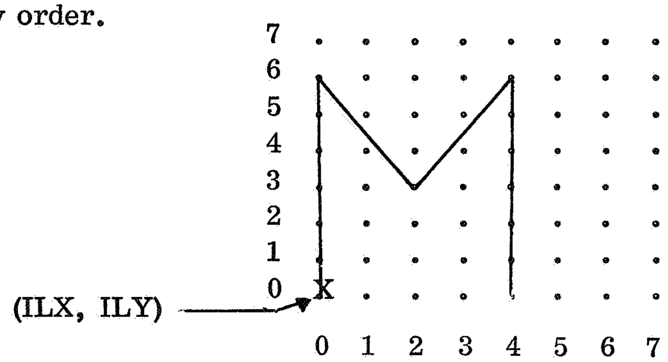


The instructions for this pattern could be read in as the single <u>octal</u> word:

PATRN1

0	4	6	6	2	2	0	6	0	0	0	0
W1	W2	H1	H2	W3	W4	H3	H4	(End of pattern)			

The pattern for M requires more than one word. It could be read in as an octal array, with a dimension of 2, in order to appear in storage in proper FORTRAN array order.



The instructions for this pattern, in octal are:

PATRN2(1)

0	0	0	6	0	2	6	3	2	4	3	6
W1	W2	H1	H2	W3	W4	H3	H4	W5	W6	H5	H6

PATRN2(2)

4	4	0	6	0	0	0	0	0	0	0	0
W7	W8	H7	H8	(End of Pattern)							

SINGLE PATTERN FOR VCHARV

A special form of VCHARV allows a single pattern to be furnished as a FORTRAN array. The following call statement may be used:

CALL VCHARV (K, NT, ILX, ILY, 0, A, 0)

A — A one-dimensional array in which the pattern must be stored. The fifth and seventh arguments must be zero. The remainder of the arguments are as described earlier for VCHARV. "A" must <u>not</u> be named on an "F" card or in an "External" definition.

TABLE STRUCTURE

When the programmer requires special characters which are not in the system he may build his own table. The following is a description which is FORTRAN II and FAP oriented. Similar procedures are applicable for the SCORS IV and the Lockheed packages. In the FAP language he can build a table similar to the tables in Figure 2-44. As with these tables, he must use an "F" card carrying the table name. Since the number of words in each pattern varies, a list of entries is employed in the search for a specific character pattern. The location TABL_V+1 must contain, in its address, the location of the list entries. Each table is compiled as if it were a FAP subprogram for FORTRAN use. Storage in FORTRAN-array type, i.e., descending storage for the table and for each individual character ends with 12 zero. For the statement VFD N/M N/M, etc., N is the number of bits allotted to M, which in turn is the quantity to be used for W (or H). (The patterns illustrated are the numeric characters 0 through 5.)

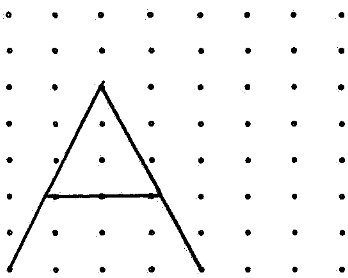
```
*         FAP
          COUNT    25
          ENTRY    TABL9V
          VFD      9/12, 9/14, 18/0
ENTR9V    VFD      9/0, 9/3, 9/5, 9/8
          VFD      3/0, 3/0, 3/4, 3/6, 3/0, 3/4, 3/6, 3/6, 12/0
          VFD      3/4, 3/4, 3/1, 3/3, 3/4, 3/3, 3/3, 3/4, 3/3, 3/0, 3/4, 3/4
N5        VFD      3/0, 3/1, 3/1, 3/0, 3/1, 3/3, 6/0, 3/3, 3/4, 3/0, 3/1
          VFD      3/0, 3/4, 3/2, 3/2, 24/0
N4        VFD      3/2, 3/4, 6/0, 3/3, 3/3, 3/0, 3/6, 3/3, 3/0, 3/6, 3/2
          VFD      3/3, 3/1, 3/6, 3/6, 3/1, 3/0, 3/6, 12/0
          VFD      3/3, 3/4, 3/3, 3/4, 3/4, 3/4, 3/4, 3/5, 3/4, 3/3, 3/5, 3/6
          VFD      3/4, 3/4, 3/1, 3/2, 3/4, 3/3, 3/2, 3/3, 3/3, 3/2, 3/3, 3/3
N3        VFD      3/0, 3/1, 3/1, 3/0, 3/1, 3/3, 6/0, 3/3, 3/4, 3/0, 3/1
          PZE      0
          VFD      3/4, 3/4, 3/5, 3/4, 3/4, 3/0, 3/4, 3/0, 3/0, 3/4, 6/0
N2        VFD      3/0, 3/1, 3/5, 3/6, 3/1, 3/3, 3/6, 3/6, 3/3, 3/4, 3/6, 3/5
          PZE      0
N1        VFD      3/1, 3/3, 6/0, 3/2, 3/2, 3/0, 3/6, 3/1, 3/2, 3/5, 3/6
          VFD      3/3, 3/2, 3/0, 3/0, 3/2, 3/1, 3/0, 3/1, 12/0
          VFD      3/3, 3/4, 3/6, 3/5, 3/4, 3/4, 3/5, 3/1, 3/4, 3/3, 3/1, 3/0
TABL9V    VFD      3/1, 3/1, 3/1, 3/5, 3/1, 3/2, 3/5, 3/6, 3/2, 3/3, 3/6, 3/6
          PZE      ENTR9V
          END
```

CHARACTER ORIENTATION AND PLACEMENT: RITE2V

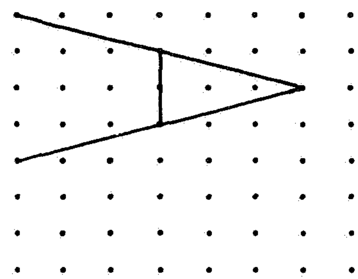
If only one character is to be displayed, VCHARV is suitable, but for several characters, RITE2V can be employed more easily. Since RITE2V uses VCHARV to generate characters, much of the preceding discussion applies. However, unlike VCHARV, which uses the lower left corner of the lattice as its position reference point, RITE2V uses the center of the 5 x 7 portion of the 8 x 8 lattice.

The following diagrams show the relation between the character lattice as it is used in RITE2V and in VCHARV. The 5 x 7 lattice is shown superimposed upon the maximum 8 x 8 lattice. The asterisk indicates the reference point (IX, IY) as used by RITE2V, while the X indicates the reference point (ILX, ILY) as used by VCHARV.

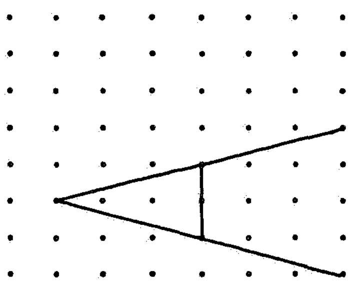
K = 90



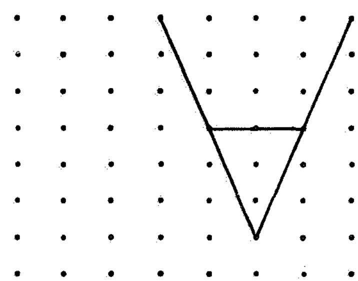
K = 0 or 360



K = 180



K = 270



Since the center is used as the character reference point by RITE2V, and since a 5 x 7 lattice is normally used, the restrictions upon the position coordinates differ slightly from those shown in the table for VCHARV. The restrictions are:

K	IX	IY
90	$2*LVW \le IX \le 1023 - 2*LVW$	$3*LVH \le IY \le 1023 - 3*LVH$
180	$3*LVH \le IX \le 1023 - 3*LVH$	$2*LVW \le IY \le 1023 - 2*LVW$
270	$2*LVW \le IX \le 1023 - 2*LVW$	$3*LVH \le IY \le 1023 - 3*LVH$
0 (or 360)	$3*LVH \le IX \le 1023 - 3*LVH$	$2*LVW \le IY \le 1023 - 2*LVW$

(One more LVH increment must be considered for the $ sign.)

The character space size and row size to be used by RITE2V depend upon the character size and the space required to prevent overlapping of characters. Since the characters in the tables normally used by RITE2V are based on a 5 x 7 lattice, the character size is, usually, 4*LVW by 6*LVH. (For the $ sign, it is 4*LVW by 7*LVH.) Also, since LVW = LVH = 3, the character size is 12 by 18, unless changed by CHSIZV. The standard quantities ISPACE and IROW used by RITE2V are 18 and 24, respectively, allowing sufficient space to prevent overlapping of characters.

The description of RITSTV shows how the programmer may redefine ISPACE, IROW, and also may name a table other than TABL1V which is normally used. The table named may be one of the alternate ones provided by the system or one of the programmer's own design.

<u>SINGLE PATTERN FOR RITE2V</u>

An alternate form of RITSTV allows the programmer to furnish a single character pattern in an array. The call statement is:

```
CALL RITSTV (ISPACE, IROW, A, O)
```

A	A FORTRAN array in which the programmer has stored one character pattern. "A" must <u>not</u> be named on an "F" card or in an "External" definition.

The last argument must be zero, to indicate to the subprogram that A is not a table. ISPACE and IROW are as defined as before.

VCHARV can also be employed to generate a single character from a pattern supplied as an array. Although VCHARV is more economical of machine time, the programmer may find it less awkward to use RITSTV to insert a character into printed information being written by RITE2V.

H. SPECIAL PRINTING

LINE PRINTER SIMULATION ON THE S-C 4020: SCOUTV

Line printing can be simulated on the S-C 4020 by means of the SCOUTV system. Whenever the statement

CALL SCOUTV

appears in the program using SCORS II and the GSFC FMS system, WRITE OUTPUT TAPE 7, will be recognized as a command to produce printed output on the S-C 4020. Using SCORS IV and the GSFC IBSYS system WRITE (7, fmt) is the appropriate statement for S-C 4020 printed output. When the Lockheed package is being used with the GSFC EXEC II system either WRITE (16, fmt) or WRITE OUTPUT TAPE 16 (the former is preferable) will be recognized as a command to produce S-C 4020 printed output. The output will be determined by the list of the write statements in conjunction with the FORMAT statements; S-C 4020 output will be obtained instead of output from the printer.

One natural application of the SCOUTV mode is the production of printed output on film for long-term storage, making it unnecessary to keep large stacks of printed paper. The use of SCOUTV as a printer simulator is discussed on the following pages. Another application is the combination of plotting and printing on one film frame.

CHANGING LOGICAL UNIT FOR SCOUTV: LUNITV

The use of SCOUTV causes the simulation of line printing on the S-C 4020 for FORTRAN logical unit n, where n is the logical unit 7 unless otherwise specified by a call on LUNITV. NOTE: n can have only one value at a time; therefore, SCOUTV can effectively be turned off for one logical unit by calling LUNITV again specifying a different unit.

The calling sequence is

CALL LUNITV (N)

where N — FORTRAN logical for which printer will be simulated on the S-C 4020.

The subroutine LUNITV is contained only in the Lockheed package for the UNIVAC 1107/1108.

BASIC SPECIFICATIONS

The following list explains most of the differences and likenesses that prevail between ordinary printing and SCOUTV printing, when there have been no programmed alterations to change the normal SCOUTV output.

1. The camera must have been selected by a CALL CAMRAV (N) statement.

2. A CALL SCOUTV statement is necessary to make printer simulation effective. The SCOUTV mode will not carry over from one Chain link or Overlay to the next; hence a new CALL SCOUTV statement is necessary in each link or overlay.

3. The first character of each BCD unit record is recognized as a carriage control character as follows:

+	No space before printing
Blank	Single space before printing
0	Double space before printing
-	Triple space before printing
1	Frame advance (equivalent to sheet eject)

An incorrect carriage control character is <u>much more critical</u> on the S-C 4020 than on a standard printer. The S-C 4020 operator cannot override the action of these characters. Non-standard characters, or loops involving a "1" can cause a great waste of film.

4. The position of print line 1 will be on the Y raster coordinate that is 23 counts from the top of the frame.

5. In the basic SCOUTV mode, it is usually desirable to advance the film as you would eject a sheet of paper, by a carriage control character "1". The first information to be printed on the new frame will then start on print line 1. This is also true if the frame is changed as a result of the line count being exceeded.

6. If the frame is changed by a CALL FRAMEV, the first carriage control character of the first FORMAT statement will affect the position of the information to be displayed in one of the following ways:

A blank will cause printing to start on print line 1.

A "0" will cause printing to start on print line 2 (46 counts from the top).

A "-" will cause printing to start on print line 3.

A "1" will cause the frame to eject, leaving a wasted frame.

Usually "+" will cause printing on line 0, right at the top of the page. This feature is not recommended.

7. Although the S-C 4020 printing will correspond line-for-line and page-for-page with printer output, the proportions of length-to-width will not be the same.

8. The spacing between lines is not governed by the typewriter mode: each new line will be started 23 raster counts below the previous line, which is more generous than the spacing provided by the typewriter mode.

9. There are 44 lines permitted per page. An attempt to write beyond the 44th line will cause the film to be advanced. Printing will continue on the new frame; the first line will be 23 raster counts from the top. The carriage control character for that line will have no effect, just as in printed output.

10. For SCORS II a WRITE OUTPUT TAPE 7 is used instead of a WRITE OUTPUT TAPE 3. For SCORS IV a WRITE (7, fmt) should be used instead of WRITE (3, fmt). The normal output on the 1107/1108 is via PRINT

statements and these should be changed to either WRITE (16, fmt) or WRITE OUTPUT TAPE 16. The FORMAT statements and list requirements are the same. Output will correspond line-for-line and page-for-page with the printed output.

11. In some cases, it may be desireable to use a WRITE OUTPUT TAPE N or a WRITE (N, fmt) statement, reading N in as a piece of data. At execution time if N = 3, normal printing will take place; if N = 7, S-C 4020 printing will result.

12. Each line can contain up to 120 characters. The first character will be positioned with its center at IX = 24, and the last one will be placed at IX = 976.

13. Once a line (of 120 characters or less) has been established, it is displayed by the S-C 4020 typewriter mode. The accidental appearance of the octal codes 12, 52, or 56 in the BCD text could cause an imperfect picture, since these codes have special uses in the typewriter mode.

Figures 2-45 and 2-46 show an exercise in the use of SCOUTV printer simulation, demonstrating some of the carriage controls, Hollerith character printing, integer, fixed, and floating point output, etc. Part of the exercise is a line count, including a count of lines skipped. Note that when a line count of 44 was exceeded, a new frame was begun.

SCOUTV VARIATIONS

The basic SCOUTV system treats a frame as if it were segmented into 44 print lines, starting on the 23rd raster count from the top of the frame and ending on the 1012th raster count from the top. Each print line is 23 raster counts deep. Printing begins on the first print line, and each line starts on the 24th raster count from the left of the frame.

The following discussion explains how these basic specifications can be altered in a number of ways.

Altering the Starting Line

Printing may be started on a print line other than the first by using the statement:

CALL SCOUTV (LINE1)

LINE1 — An integer which specifies the print line on which the next printing in the SCOUTV mode will begin.

For example, if the statement CALL SCOUTV (5) is used, the next WRITE OUTPUT TAPE statement will place printing on the 5th print line (the raster position that is 5*23 counts from the top of the frame). The carriage control character (even the "1" for "eject") will be ignored for that line.

For LINE1 to be in force, this statement must be executed in advance of the associated WRITE OUTPUT TAPE statement.

If no other alterations are made, LINE1 may have any value from 1 to 44, and the maximum number of lines of printing will be (45 - LINE1).

Since LINE1 will be effective only for one frame, CALL SCOUTV (LINE1) should be restated if printing is to start on the named LINE1 on the next frame.

Altering the Starting Line and Line Spacing Increments

Further alteration of the basic SCOUTV specifications may be made by using the statement:

CALL SCOUTV (LINE1, INC)

LINE1 — An integer which specifies the print line on which the next printing in the SCOUTV mode will begin. The actual position of the first line on the frame will depend upon the value of the second argument, INC.

INC — An increment of line spacing. The frame will be considered to be segmented into print lines that are INC

```
                                                                                  0000-00
                                                                                  033   033
CARRIAGE CONTROL CHARACTER IS BLANK   THIS IS LINE   1        1.45       0.14+01

CARRIAGE CONTROL CHARACTER IS   0     THIS IS LINE   3        3.45       0.34+01


CARRIAGE CONTROL CHARACTER IS   -     THIS IS LINE   6        6.45       0.64+01




CARRIAGE CONTROL CHARACTER IS   4     THIS IS LINE  10       10.45       0.10+02
CARRIAGE CONTROL CHARACTER IS BLANK   THIS IS LINE  11       11.45       0.11+02

CARRIAGE CONTROL CHARACTER IS   0     THIS IS LINE  13       13.45       0.13+02


CARRIAGE CONTROL CHARACTER IS   -     THIS IS LINE  16       16.45       0.16+02




CARRIAGE CONTROL CHARACTER IS   4     THIS IS LINE  20       20.45       0.20+02
CARRIAGE CONTROL CHARACTER IS BLANK   THIS IS LINE  21       21.45       0.21+02

CARRIAGE CONTROL CHARACTER IS   0     THIS IS LINE  23       23.45       0.23+02


CARRIAGE CONTROL CHARACTER IS   -     THIS IS LINE  26       26.45       0.26+02




CARRIAGE CONTROL CHARACTER IS   4     THIS IS LINE  30       30.45       0.30+02
CARRIAGE CONTROL CHARACTER IS BLANK   THIS IS LINE  31       31.45       0.31+02

CARRIAGE CONTROL CHARACTER IS   0     THIS IS LINE  33       33.45       0.33+02


CARRIAGE CONTROL CHARACTER IS   -     THIS IS LINE  36       36.45       0.36+02




CARRIAGE CONTROL CHARACTER IS   4     THIS IS LINE  40       40.45       0.40+02
CARRIAGE CONTROL CHARACTER IS BLANK   THIS IS LINE  41       41.45       0.41+02

CARRIAGE CONTROL CHARACTER IS   0     THIS IS LINE  43       43.45       0.43+02
```

(NOTE: Carriage Control Characters 4-8 are not to be used at GSFC)

Figure 2-45

```
                                                                                      0000-00
                                                                                      034   034
CARRIAGE CONTROL CHARACTER IS  -    THIS IS LINE  46        46.45       0.46+02


CARRIAGE CONTROL CHARACTER IS  4    THIS IS LINE  50        50.45       0.50+02
```

(NOTE: Carriage Control Characters 4-8 are not to be used at GSFC)

Figure 2-46

raster counts deep. The maximum number of print lines will be 1012/INC, since the basic limit for the bottom line is 1012 raster counts below the top of the frame.

The standard LINE1 and INC for SCOUTV are reset by a CALL SCOUTV (1, 23).

The carriage control character for the first line of information to be printed following this call statement will be ignored. (Note that this also applies to the character for "eject.") Therefore, printing will start on the raster coordinate that is LINE1*INC raster counts from the top of the frame.

INC will remain in effect for subsequent SCOUTV mode printing in the program until altered by another CALL SCOUTV (LINE1, INC).

For LINE2 or INC to be effective, the CALL SCOUTV (LINE1, INC), statement must be executed before the associated WRITE OUTPUT TAPE statement.

NOTE: A CALL SCOUTV (with or without arguments) placed anywhere in a program (or chain link) will cause the SCOUTV mode to be used for all WRITE OUTPUT TAPE statements in the program or chain link, even those which precede the CALL SCOUTV. That is, the basic SCOUTV mode will be in effect until a CALL SCOUTV with arguments is actually executed.

Figure 2-47 illustrates the use of SCOUTV with arguments that alter the basic specifications. Note the use of INC = 16, which caused the lines to be closer than normal, enabling the same 50 lines shown on Figures 2-45 and 2-46 to be printed on one frame. Since LINE1 = 4 and INC = 16, the first line displayed is actually 4*16 raster counts below the top of the frame.

PRECISE CONTROL OF RASTER POSITIONS FOR SCOUTV PRINTING: LOCSTV

LOCSTV is available only in the SCORS II and SCORS IV packages. In the preceding discussion, SCOUTV was presented as a means for simulating printer output on the S C 4020. For such simulation, vertical positioning is

```
CARRIAGE CONTROL CHARACTER IS BLANK   THIS IS LINE  1        1.45       0.14+01

CARRIAGE CONTROL CHARACTER IS   0     THIS IS LINE  3        3.45       0.34+01


CARRIAGE CONTROL CHARACTER IS   -     THIS IS LINE  6        6.45       0.64+01



CARRIAGE CONTROL CHARACTER IS   4     THIS IS LINE 10       10.45       0.10+02
CARRIAGE CONTROL CHARACTER IS BLANK   THIS IS LINE 11       11.45       0.11+02

CARRIAGE CONTROL CHARACTER IS   0     THIS IS LINE 13       13.45       0.13+02


CARRIAGE CONTROL CHARACTER IS   -     THIS IS LINE 16       16.45       0.16+02



CARRIAGE CONTROL CHARACTER IS   4     THIS IS LINE 20       20.45       0.20+02
CARRIAGE CONTROL CHARACTER IS BLANK   THIS IS LINE 21       21.45       0.21+02

CARRIAGE CONTROL CHARACTER IS   0     THIS IS LINE 23       23.45       0.23+02


CARRIAGE CONTROL CHARACTER IS   -     THIS IS LINE 26       26.45       0.26+02



CARRIAGE CONTROL CHARACTER IS   4     THIS IS LINE 30       30.45       0.30+02
CARRIAGE CONTROL CHARACTER IS BLANK   THIS IS LINE 31       31.45       0.31+02

CARRIAGE CONTROL CHARACTER IS   0     THIS IS LINE 33       33.45       0.33+02


CARRIAGE CONTROL CHARACTER IS   -     THIS IS LINE 36       36.45       0.36+02



CARRIAGE CONTROL CHARACTER IS   4     THIS IS LINE 40       40.45       0.40+02
CARRIAGE CONTROL CHARACTER IS BLANK   THIS IS LINE 41       41.45       0.41+02

CARRIAGE CONTROL CHARACTER IS   0     THIS IS LINE 43       43.45       0.43+02


CARRIAGE CONTROL CHARACTER IS   -     THIS IS LINE 46       46.45       0.46+02



CARRIAGE CONTROL CHARACTER IS   4     THIS IS LINE 50       50.45       0.50+02
```

(NOTE: Carriage Control Characters 4-8 are not to be used at GSFC)

Figure 2-47

controlled by a line count and the increment between lines. Lateral positioning is dependent upon the built-in left margin and the number of character spaces provided by the associated FORMAT statement.

The range of SCOUTV applications has been greatly extended in SCORS II and SCORS IV only, by means of routines which permit precise control of raster position, with the printing still under FORMAT control. Two routines, LOCSTV and LOCSAV, which permit setting and retrieval of raster positions add the following capabilities to the SCOUTV system.

1. Printing can be done below IY = 11. (The basic SCOUTV does not permit writing below the 1012th raster position, counted from the top of the frame.)

2. Printing can be done to the left of IX = 24. (This area is inaccessible to the basic SCOUTV system, which uses a fixed left margin of IX = 24.)

3. Printing can be started at a point which is independent of a line count, line increment, or number of character spaces. (For the basic system, which simulates line printing, printing can only be positioned in units of character widths and of print line depth.)

4. Printing can extend to the right of IX = 976, by starting the line at IX $\geq$ 25 (this area is inaccessible to the basic system, since the maximum (120 characters) SCOUTV line starts at IX = 24 in the basic SCOUTV, and thus ends at IX = 976.)

The LOCSTV provides these facilities by means of the call statement:

CALL LOCSTV (IX, IY, IB)

IX — The X-raster position to be used as the starting point of the next line of printing. Stated in another way, IX is the new left margin, which will be effective for subsequent lines and frames, until another CALL LOCSTV is encountered.

The programmer must ensure that information to be printed on subsequent lines can be accommodated between IX

EXERCISE IN THE USE OF LOCSTV WITH ACOUTV IN SPECIAL PRINTING APPLICATIONS

ENGINEER J. MATTEL
DEPT.-GROUP 282-072
EXT. 2651
JOB. NO. 230-30

PROJECT SABRE - SUB
PART PERISCOPE
MATERIAL PLASTIC

MODEL NO. C 8 IN
TEST NO. 33T
DATE 10/ 5

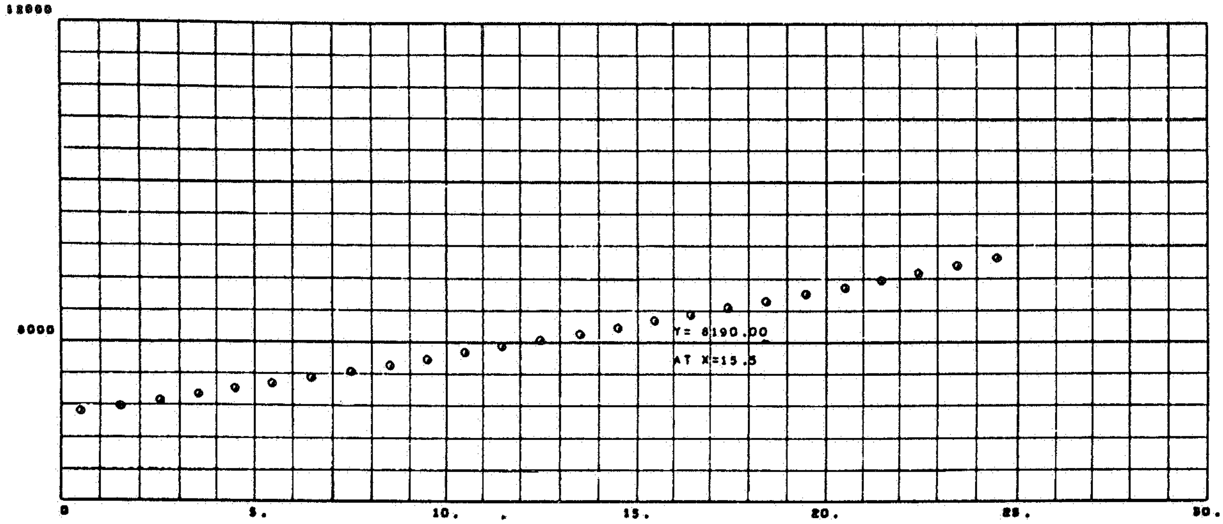
7125.00	7488.45	7870.43	8271.90	8693.85
7196.25	7563.33	7949.14	8354.62	8780.79
7268.21	7638.96	8028.63	8438.17	8868.60
7340.89	7715.35	8108.91	8522.55	8957.29
7414.30	7792.51	8190.00	8607.78	9046.86

Figure 2-48

GRAPH AND PRINTED INFORMATION COMBINED ON ONE FRAME

NOTE..THIS EXERCISE SHOWS FURTHER WAYS IN WHICH SCOUTV PRINTING AND LOCSTV MAY BE USED

0000-00
057 05



7125.00	7488.45	7870.43	8271.90	8693.85
7196.25	7563.33	7949.14	8354.62	8780.79
7268.21	7638.96	8028.63	8438.17	8868.60
7340.89	7715.35	8108.91	8522.55	8957.29
7414.30	7792.51	8190.00	8607.78	9046.86

Figure 2-49

and the right edge of the frame. IX should be specified such that

$$1023 - IX \geq 8 * (\text{number of characters in the line to be printed})$$

IY — A Y-raster position counted from the BOTTOM of the frame, to be used as a reference for subsequent lines of printing. Since the position IX, IY will be referenced as if it were the starting point of the previous line of printing, the next line will start at IX, IY only if the carriage control character + (to suppress spacing) is used.

The position of subsequent lines of SCOUTV mode printing relative to IY will depend upon the carriage control characters used and upon the value of the current line increment, INC.

IY is effective only for subsequent lines on the same frame It will be changed (reset to the top of the frame) if the frame is advanced, or if one of the forms of CALL SCOUTV intervenes between a CALL LOCSTV and the affected WRITE OUTPUT TAPE statement.

IB — Limit below which printing will not take place, expressed in raster counts from the BOTTOM of the frame. It will be effective for subsequent lines and frames until another CALL LOCSTV is encountered.

To return these indicators to the basic SCOUTV specifications, use CALL LOCSTV (24, 1000, 11).

Note particularly that, unlike IX and IB, the IY reference point will not carry over from frame to frame, nor will it survive execution of a CALL SCOUTV. If the IY reference is to be used on a new frame, the CALL LOCSTV statement should be repeated.

One use of LOCSTV is illustrated in Figure 2-48.

An illustration of how SCOUTV mode printing can be combined with a graph is shown in Figure 2-49. That example shows also another way to print tabular data using SCOUTV and LOCSTV.

PRESERVATION (OR RETRIEVAL) OF RASTER POSITIONS USED BY SCOUTV:

LOCSAV - LOCSAV is only available in the SCORS II and SCORS IV packages.

The raster coordinates of the starting point of the line most recently printed in the SCOUTV mode can be stored in ISX, ISY by using the statement:

```
CALL LOCSAV (ISX, ISY)
```

The programmer can use this information to relate other types of S-C 4020 output to the location of the last line of print.

LOCSAV will also aid in a special situation that occurs with interrupted printing. If SCOUTV printing is interrupted on one camera in order to display one or more frames on the alternate camera, the SCOUTV printing and film advance for the alternate camera will alter the line reference information. On return to the original frame on the first camera, the programmer will be unable to resume SCOUTV printing at the next line unless he uses LOCSAV and LOCSTV. The following sequence shows the procedure that might be utilized.

```
.
.
.
CALL SCOUTV
.
.
.
CALL CAMRAV (35)
.
CALL FRAMEV
.
.
.    Write a portion of this 35mm frame, using the appropriate
     WRITE OUTPUT TAPE and FORMAT statements.
```

```
        .
        .
        .
CALL LOCSTV (M1X, M1Y, 11)
CALL CAMRAV (35)
        .
        .
        .
     Continue writing on 35mm film as if no interruption had taken place.
        .
        .
```

I. PRINTING SUBROUTINE: VCPS

Subroutine VCPS is available only in the SCORS II and IV packages.

This subroutine produces a tape for printing matrix symbols, vector characters and vector lines. Clues which are coded in the keypunch data may provide for automatic line justification, column printing and page advances - all done automatically when column or page ends occur. Column graphs may be drawn and data may also provide for precise positioning of lines and characters, without the automatic justification feature.

VCPS operates by means of calls to particular subroutines of the S-C 4020 package being used, and is entered by the call statement:

```
CALL VCPS
```

HEAD CARD

Maximum use of the automatic features occurs when a blank card heads data. An S punched in column 71 of this card causes leftward shift for lower case characters in some instances to produce more attractive print. Also, the S sets the condition for a search for usable format information when improper format cards (described later) are encountered. Page numbers print automatically at the lower left.

The card which must occur next is called a format card.

FORMAT CARD

This card may have information in card columns 1-9 (balance not read).

Columns 1-3 show the number of characters, spaces included, to a line. The maximum value allowed is 155, but the useful limit is 128. For more than 128 characters use a frame limit card without the leading blank card. The frame limits card is explained later on. The minimum character spacing is set automatically to 8 raster points.

Columns 4-5 choose matrix or vector characters, set character and line spacing and set character size for vector characters. A blank column 4 sets matrix point, while a value in column 4 sets IW for CHSIZV. The increment for spacing will be (4 x IW) + 3, but no less than 8. Column 5 sets IH for CHSIZV and line spacing for both matrix and vector characters. A blank column 5 sets lines 20 raster counts apart. When column 5 is not a blank or a zero, the formula is (6 x IH) + (4 x IH) for increments, with the lower and upper margins at 36 and 24 raster points. The margins will increase for large values in column 5, leaving fewer lines.

> NOTE: When column 5 value is a 5 or greater, characters are called from TAB11V (upper case) and TAB15V (lower case) which are block letters and not suitable for use in a smaller width.

Column 6 sets the number of columns. A blank, zero or 1 punch gives a page with one centered column. Multiple columns are centered, with a three character distance between columns. There will be a return from VCPS if the size and number are excessive. The comment NXV ERROR TOO NEAR FRAME EDGE will print. As many as nine columns may be attempted.

To suppress return on this type error, punch S in column 71 of a blank card used before a format card to induce fullest use of automatic features. VCPS will attempt to use the format by reducing character size and number of columns. If these efforts fail, there is a return to the calling program.

The condition set by the S also produces character adjustment. Symbols occurring after f, i, j, l and t will be adjusted leftward for more attractive print. The characters used are in TAB05V and TAB07V, but adjustment only occurs for lower case characters.

When the last word of a line is clued for all caps, an asterisk should be entered following the last character.

To produce unequal left and right margins, columns 13-18 and 19-24 of the leading otherwise blank, card may be used. This card is actually a frame limits card. Integer values must be used for raster values of margins. For the left margin, enter raster count in columns 13-18 and for right margin use columns 19-24.

Column 7 provides for the suppression of a data listing by means of a 2 punch.

Column 8 may carry a SKIP or HOLD character. For a blank column 8, the equal sign is set up. (See COLUMN TEXT CARD description which follows.)

Column 9 sets the character intensity. A blank indicates an intensity of 1. The intensity is the column 9 value plus 1 and the greatest acceptable intensity to VCPS is 3.

COLUMN TEXT CARDS

The column text cards carry the texts for printing. The number of columns read will be one greater than indicated in columns 1-3 of the format card. VCPS reads columns 1-72 of text cards until the format card requirement is met. For a character count of 72, two cards will be read - 72 columns from one card and one column on the succeeding card. In lines of 144 characters, three cards would be read for each line of print. The additional column read each time is called the "KEY" column and should never contain any information other than one of the KEY column numbers.

KEY column numbers are: 1, 3, 4, 5, 6, and 7

KEY 1	Directs VCPS to expect column vector line cards.
KEY 3	Gives frame advance and column text card must follow.
KEY 4	Provides for reading a frame limits card as the following card. By using KEY 4, sections of VCPS other than the automatic printer section may be entered without a frame advance.
KEY 5	Gives a return from VCPS.
KEY 6	Provides for reading a new format card, whereby any format parameters may be changed. Printing continues on the same frame if there is room for at least one line of print, otherwise frame advances.
KEY 7	Operates as KEY 6, but the frame <u>always</u> advances with printing starting at the top of the new page.

SKIP or HOLD mark may be any character in column 8 of the format card. When the selected character occurs singly in the text, subsequent characters in the line do not print and the remaining text will justify. Thus, when a word near line-end is begun and does not effectively hyphenate, placing a SKIP mark ahead will delete it. Another use is to save time by preventing execution of blanks near a line-end.

Blank cards are used to represent line skips and must occur in pairs (or multiples) when more than one card is used per line. Cards for skips should have a SKIP mark in column 1 to shorten the execution time, and a digit punched in column which causes a skip of that many lines.

> NOTE: When column 1 has a SKIP mark, any character in column 2 is interpreted as a decimal code count for line skipping.

When column 8 on a format card is blank, the equal mark is set as a SKIP or HOLD mark, and does not print. Of course, a character must be chosen which does not occur within the text. A new character may be selected at any time by using KEY 6 on a column text card to afford reading a new format card.

CONVENTIONS

Characters used for CLUES when doing Column Printing are:

> . * +) (/ and SKIP or HOLD mark, as indicated by column 8 of the effective format card.

An S in column 71 of a leading card causes use of TAB05V or TAB07V tables constructed for use when the 'character squeeze' justification feature is operative.

Clues are effective only within one print line.

When printing Vector Characters, TABL5V, (a table of lower case characters) is used unless a CAPITALIZING or ITALICIZING clue occurs.

A capital prints when a symbol occurs after 2 or more spaces, or directly after a comma, or when an asterisk has occurred next to a symbol. Numerals, the plus, and the slash must be treated as capitals.

When a character has 3 or more spaces before it and yet must be printed lower case, enter a period immediately before the character to prevent capitalization. The period thus used is treated as a clue and will not print.

As general practice it is best to use blanks on both sides of a , * +) (or / when it must be printed. The comma is an exception. The comma prints when it occurs just before another comma or just before an asterisk (*), and <u>always prints when in column 2.</u> When capitals have been 'turned on' by the asterisk, commas cease to be 'clues' and will print.

The asterisk used as a clue will induce exclusive use of TABL1V, (or TABL6V in Italics) until another clue asterisk occurs.

Two asterisks occurring between letters or numerals cause shift into or out of capitals. Spaces of the asterisks are used to print the balance of the symbols in the word so that no gap is left.

The Plus (+) as a clue at first encounter induces the use of Italics. Characters will be from TABL6V or TABL7V depending on the capitalization clues, until another plus occurs.

Two plus signs between letters or numbers cause shift into or from Italics. Spaces occupied by the plus signs are closed so that no gap will appear in the print.

Any number of characters in one line will be underlined from any symbol with a <u>right parenthesis</u> just before it to a symbol having a <u>left parenthesis</u> immediately after it. Greek symbols are set by a Left Parenthesis placed immediately after a character when an underline has not been begun. Greek symbols will be 'turned off' at the next occurrence of the condition.

A slash used as a clue will increase or decrease character intensity. Intensity increases by 1, and another clue / decreases intensity by 1. VCPS will not use intensity greater than 3.

For matrix print, only 3 clue devices are operative. Underlining may be done with 'inverted parenthesis' as described above, and SKIP or HOLD marks are valid. The asterisk (*) will produce brighter intensity when used as a clue. All other "CLUES" print.

Clue marks become spaces when used as clues. Successive clue marks may produce capitals, since a symbol appearing after two spaces is capitalized. Double asterisks or plus signs within words are exceptions for this rule. See use of double asterisks and plus signs above.

INDENTATION

Text will justify to left when first print character is in column 1, 2, or 3. Indentation is held when first print symbol is beyond column 3.

COLUMN GRAPHS

When 1 is found in Key Column of a Column text card, VCPS expects data for Vector line drawing within a column width. Following data card must have integer values thus:

Columns	
1-4	Character count at vector line start.
5-8	Line count from previous print line to vector start.
9-12	Character count at vector line end.
13-16	Line count from previous print line to vector end.
17	Any integer directs VCPS to expect column text card next.

Printing resumes at line below last text line, to provide for labels on the vector lines. To resume printing below the drawn vectors, use a line skip card (a card having SKIP mark in column 1.) See information on use of column 2 of a SKIP card for producing multiple skipping. The information is found in section on use of Clue Marks.

When VCPS is supplied data as described, margins, line spacing, and increments between characters are computed by the Subroutine. For more precise control of margins and spacing a FRAME LIMITS CARD may precede data, and the combination of blank card plus format card is not used. When column text cards follow a frame limits card having applicable columns punched as described below, automatic column and page advance features operate but framing and spacing will be as set on the frame limits card.

FRAME LIMITS CARD

A frame limits card precedes any data for execution by VCPS. When such a card is blank, the subroutine expects a format card for automatic frame determination. A card for precise frame control will have information that controls spacing absolutely. Vector and matrix characters will print 1 unit apart in X, and justification may increase spacing between words. It is helpful to set the frame geometry so that 1023 is an integral multiple of a frame unit in raster points.

Frame Limits Card Format

Column	Information
1-6	Floating X value at left frame edge
7-12	Floating X value, right frame edge
13-18	Integer, raster count, left margin
19-24	Integer, raster count, right margin
25-30	Floating Y value, lower frame edge
31-36	Floating Y value, top frame edge
37-42	Integer, raster count, lower margin
43-48	Integer, raster count, top margin
49-54	Floating X value, column one start

55-60	Floating Y value, first line start
61-63	Integer is character count per line
64	Integer, sets character width, IW
65	Integer, sets character height, IH
66-67	Integer, sets table of characters 1 for TABL1V, 5 for TABL5V, Etc.
68	Integer, sets character intensity Subroutine adds 1 to value entered.
69	Not read.
70	Key integer controls reading and processing of following cards. Blank or 0 when text card follows. Note: Column 70 is not operative for integer values when columns 61-63 are non-blank. When column text cards are not next, columns 61-63 must be blank, and key numbers are effective in column 70. Blank or 0 if text card is next; 1 if vector line card is next; and 2 if decimal card follows.
71	Key punch character will become SKIP or HOLD mark. Blank column 71 results in use of equals as SKIP or HOLD.
72	Not operative for printing from column text cards. When text cards do follow integer controls orientation of vector characters and print lines. Blank or 0 for upright print; 1 to rotate print 90 deg. CCW; 2 to rotate print upside down; and 3 to rotate print 90 deg. CW.

TEXT CARD FORMAT

A text card may carry information for matrix or vector character printer. Control of position is precise. Automatic columns and page advances do not happen, and lines do not justify. Each line will print exactly along the Y value on the card. This type of card is expected after a Frame Limits Card which contains no information beyond column 48. No clue marks (as described for column text cards) have any "Clue" effects when found in a text card.

Column	Information
1-3	Integer sets character width, INCRW values must not exceed 7. Blanks or zeroes, give matrix print,
4-6	Integer sets character height, INCRH value must not exceed 7. For "slanted" matrix print, value is raster x distance between symbols. Note: Vector character size varies slightly with table used. Maximum width or height is (8 x INCRW), or 56 raster points.
7-8	Not read.
9-11	Integer selects character table. 1, 0, or blank select TABL1V, and 2, 3, 5, t, ., 8, 9 for TABL2V, etc. For slanted matrix print, value is raster Y distance between symbols.
12	Not read.
13-18	Floating X value at center of first text character. Value should be a whole number. Decimal places are not retained in conversion to rasters. First text symbol is character in column 25, even if column 25 is blank.

	Blanks or 0's give start at raster 32 + left margin, or at 1.0 on frame. (The greater value is used.) X is given as for normal print when rotation is done (See column 72.)
19-24	Floating Y value is used to establish centerline thru a line of text. Digits to right of decimal are used to 2 places, but truncation occurs if column 72 is 1 or 3. Always give Y as for normal frame.
25-66	42 characters of text. More than 42 characters require additional cards. Starting X value on a continuing card must be 42 greater than that given on the previous card. 82 is practical limit for number of vector characters. 123 is limit for matrix symbols using text cards.
67-68	Integer value is number of times each character is struck - 1, blanks, or 0's for once. 1 for 1 additional strike, etc.
69	Not read.
70	Integer is clue for processing next card. Blank or 0 when same type card follows; 1 when a vector line card is next; 2 when a decimal code card follows; 3 to advance frame (a frame limits card must follow); and 4 to change frame geometry without a frame advance (frame limits card must follow).
71	SKIP or HOLD mark. Blank sets equals as as mark. Select character does not print.
72	Integer is clue for rotation. Blank or 0 for normal upright print;

1 to rotate print and frame CCW 90;
2 to rotate print and frame CCW 180; and
3 to rotate print and frame CCW 270.

VECTOR LINE CARDS

Column	Information
1-6	Floating X value at vector start.
7-12	Floating Y value at vector start.
13-18	Floating X value at vector end.
19-24	Floating Y value at vector end.
25-66	Not read.
67-68	Intensity integer, blank, 0 or 1 for drawing vector once, 2 for twice, etc.
69	Not read.
70	As described for text cards.
71	Not read.
72	As described for text cards.

DECIMAL CODE CARDS

Column	Information
1-24	As for text cards but columns 1-3 must not be blanks or 0's.
25-26	Decimal code integer for character wanted. (See S-C 4020 Manual)
27-29	Degree integer for character orientation. (The symbol rotates, frame does not.)

	Blanks or 90 for upright character; 180 to rotate 90 deg. CCW; 270 for an upside down symbol; and 360 for 90 deg. CW rotation.
30-66	Not read.
67-68	Intensity integer as for vector cards.
69	Not read.
70	As for text and vector cards.
71-72	Not read.

KEYPUNCH CHARACTERS FOR VCPS

TABLE: CHARACTERS FROM CARD PUNCH:

1V	ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789=.-,()+*$/"
5V	abcdefghijklmnopqrstuvwxyzœ!¡¿ñ´`¨ç^=.-,()&*?†'
6V	*ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789=.-,()+*$/"*
7V	*abcdefghijklmnopqrstuvwxyzœ!:¿ñ ˆ`¨çÇ=.-,()+*?/"*
8V	ΑΒΞΔΕ ΓΗΙΘΚΛΜΝΟΠ ΡΣΤΥΦΩΧΨΖ C –
9V	αβξδε γηιθκλμνοπ·ρστυφωχψζ
11V	**ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789=.-,()÷×"/"**
15V	**abcdefghijklmnopqrstuvwxyzo!ááíóó'?â"=.-,()÷×ø/"**

Note: Characters above are produced with

LVW = 2

LVH = 3

PRINTING WITH VCPS SUBROUTINE
(SAMPLE DATA CARDS- OUTPUT ON NEXT PAGE)
CAPITALIZING,CLUES ARE,2 OR MORE,SPACES,,THE,COMMA, AND,THE,ASTERISK.==

,DOUBLE,ASTERISKS WITHIN A WORD CAUSE CHANGE TO OR FROM,CAPITALS.,FOR TWO EXAMPLES.*JA**CKSON, NON-**E**UCLIDEAN. .==

,THE,PLUS,SIGN IS THE+,ITALICIZING+CLUE.==

,TWO,PLUS,SIGNS WITHIN A WORD CAUSE CHANGE TO OR FROM,ITALICS, AND SPACES CLOSE AS WITH DOUBLE ASTERISKS.,FOR EXAMPLE, NON-++COOPERATIVE, UN++TIED. ==

)UNDERLINING(MAY BE DONE WITH INVERTED PARENTHESES.==

,A,LEFT,PARENTHESIS WHICH IMMEDIATELY FOLLOWS A CHARACTER IS THE CLUE FOR CHANGING TO OR FROM,GREEK,CHARACTERS WHEN AN*UNDERLINE HAS NOT BEGUN.* ,FOR,EXAMPLE,,NOW DO A,GREEK,ALPHABET(ABCDEFGHIJKLMNOPQRSTUVWXYZ(ABCDEF.

,THE,EQUAL,SIGN IS A SKIP MARK, UNLESS ANOTHER SYMBOL IS SELECTED.,A LONE ,SKIP,MARK FOLLOWED BY A,BLANK CAUSES A SKIP TO NEXT LINE.,ANY SUBSEQUENT CHARACTERS DO NOT PRINT, AND THE LINE WILL BE JUSTIFIED.,TWO,SKIP,MARKS IN SUCCESSION HOLD A LINE END BY PREVENTING JUSTIFICATION OF THE RIGHT EDGE. ,FOR EXAMPLES,=
,DELETE CHARACTERS AFTER THE,SKIP,MARK. = THESE SYMBOLS WILL BE DELETED.
,DELETE CHARACTERS AFTER,TWO,SKIP,MARKS. == THESE SYMBOLS WILL BE DELETED.

Capitalizing Clues are 2 or more spaces, The Comma, and The Asterisk.

Double Asterisks within a word cause change to or from Capitals. For two examples, JAckson, non-Euclidean.

The Plus Sign is the *Italicizing* clue.

Two Plus Signs within a word cause change to or from Italics, and spaces close as with double asterisks. For example, non- *cooperative.*

Underlining may be done with inverted parentheses.

A Left Parenthesis which immediately follows a character is the clue for changing to or from Greek Characters when an UNDERLINE HAS NOT BEGUN. For Example, Now do a Greek Alphabet αβϑδε γηιθκλμν [illegible] Abcdef.

The Equal Sign is a skip mark, unless another symbol is selected. A lone Skip Mark followed by a Blank causes a skip to next line. Any subsequent characters do not print, and the line will be justified. Two Skip marks in succession hold a line-end by preventing justification of the right edge. For examples,
Delete characters after the Skip Mark.
Delete characters after Two Skip Marks.

SECTION III

CSC PLOTTER PROGRAMMING SYSTEM

CSC PLOTTER PROGRAMMING SYSTEM

The CSC Plotter Programming System links two separate machines, the S-C 4020 Microfilm Recorder (or Plotter) and the UNIVAC 1107/1108 Thin Film Computer, operating under the EXEC II monitor. The system consists of two approaches to the plotting problem: A. the Utility Subroutines and B. the Basic Subroutines themselves.

A. UTILITY SUBROUTINES

The two utility subroutines, PLOT and PLBGIN, constitute a complete plotting system in that they automatically perform most plotting tasks with a minimum of user intervention. PLBGIN, which must be called prior to a call on PLOT, constructs a temporary buffer and writes an identification frame. PLOT, a generalized plotting routine, has a variable length calling sequence and requires as few as three arguments (yielding automatic scaling and gridding to fit the data, with minimal labeling), but allows as many as twelve arguments, whereby almost all functions are in the user's control. PLOT performs such functions as storage allocation, frame advancing, construction of buffer areas, and other housekeeping functions not directly related to the preparation of data to be plotted.

B. BASIC SUBROUTINES

Basic subroutines are also offered that provide a closer control of format and data presentation, and more extensive lettering. These subroutines, except for BUFFER and SET, have fixed-length calling sequences; the user must reserve buffer areas and call subroutines BUFFER and PLOTID himself, and may generate as much data per frame as he choses. The basic subroutines are listed as follows: BUFFER, PLOTID, GRID, LETTER, LETTR2, PLOTI, PLOTP, PLOTH, PLOTL, PLOTVL, PLOTVQ, SET, PTYPE, VANCE, RITE, JECT, CAMERA, EXPAND, REDUCE and ENDFLE.

Both the Utility and the Basic Subroutines are called within FORTRAN IV coded programs.

CAPABILITIES PROVIDED BY THE CSC PLOTTER PROGRAMMING SYSTEM

A user has the option of either using the automatic gridding and coordinate scaling feature, or of specifying the grid and coordinate scaling by using the appropriate subroutine and calling sequences; the degree of character exposure can also be selected, and points can be connected by either a straight line, or a quadratic curve; he can, when doing complex calculations, store plot instructions. Finally, a user has at his disposal two different character sets, that of the S-C 4020 Typewriter, and that of Subroutines LETTER and LETTR2; each character set consists of alphanumeric, and special characters; each subroutine offers the user a specific option. LETTER can be used to cause different size character printing as well as vertical printing. LETTR2 causes the characters to be printed sideways.

As compared to the packages described in the previous section, the CSC Plotter Programming System does not provide as much of an automatic scaling capability. However, the CSC package does provide a rather complete plotting system, which features the flexibility in defining buffer areas and the capability for writing multiple plot tapes.

GENERAL OPERATING PROCEDURES

The Plotter subroutines described in this section of the manual are FORTRAN IV compatible and are available in the 1107/1108 libraries of subroutines.

TAPE ASSIGNMENTS

A. UTILITY SUBROUTINES

The user must prepare a control card to assign the magnetic tape on which the plotter subroutines will draw points and vectors. If the Utility Subroutines are used, output must be made on logical Tape Unit "A." The control card following the RUN card must be:

∇bASGba

and the 1107/1108 Operator must be instructed to "Plot A"

In addressing this tape in FORTRAN, care must be taken that 4020-type commands are written. If "A" is addressed as Unit #0 (its usual designation), the 1107/1108 I/O system will produce FORTRAN-type output; provision has been made in the I/O system that, if Unit "A" is addressed as Unit #30, control will be shunted to a part of the Plotter package.

B. BASIC SUBROUTINES

If the Basic Subroutines are used, any tape (A-H) may be used, and these units correspond to Units #40-47. This correspondence is a feature of 1107/1108 System Subroutine. NTAB$, and users having a different NTAB$ for some special purpose must be sure to duplicate these tape assignments.

All tapes used must be properly assigned by the ASG control card.

The tapes are specified as numbers corresponding to locations in NTAB$. These locations in turn must contain values in the range 120-127 corresponding to A-H, respectively. After defining the buffer area, the other subroutines may be called as needed.

In addition to these subroutines, plot instructions may be generated by the FORTRAN statement WRITE (N, FMT) LIST. The tape unit here is specified as indicated above. The line control is limited to suppress spacing (+), double-space (0), advance film (1), or single-space (all other characters). Printing may be started at any line, however, by the use of subroutine SET. The argument y in the statement, CALL SET (y), is a y coordinate in the S-C 4020 coordinate system. This value establishes a basic position, which is then modified according to the space-control character; i.e., a single space-control character would add 16 to y before printing. The "long call" of SET does not use a space-control character.

The following is a portion of NTAB$ showing logical tape numbers 40-47 designated as plot tapes. This particular arrangement makes all eight tapes available for plotting, yet puts no restrictions on their use for other purposes.

NTAB$	UNIT 0	+ "A"	
	UNIT 1	+ "B"	
	UNIT 2	+ "C"	
	⋮	⋮	
	UNIT 40	+120	Plot Tape A
	UNIT 41	+121	Plot Tape B
	⋮	⋮	⋮
	UNIT 47	+127	Plot Tape H

The programmer should note that the logical unit numbers designated as plot tapes are not available for use except as outlined in this report. However, the physical tape units may be addressed by either FORTRAN or SLEUTH references in the usual manner.

PLOTTER OPERATION

During the 1107/1108 run, Utility or Basic Subroutines are called by standard FORTRAN calling sequences. The calls generate S-C 4020 commands on tape. This tape is then transferred to the plotter, which, when activated, uses either Camera 1, 2, or 3 to plot on either 35 or 80mm. film (for microfilm or hard-copy output, respectively) or both, according to the tape instructions. The film is then removed from the plotter by the operator, and is developed into the appropriate output form.

FORM SLIDES

As stated previously, the preparation of form slides is expensive and not generally recommended. However, some slides are currently available at the GSFC and information concerning them, as well as advice concerning their preparation is contained in Appendix E.

A. UTILITY SUBROUTINES

INITIALIZATION TO PLOTTING: PLBGIN

PLBGIN constructs a temporary buffer region and writes an identification frame on Tape Unit "A." PLBGIN must be called before the first call of PLOT, and should be called only once.

PLBGIN uses an entire frame, advancing the film before returning to the master program.

The calling sequence is

CALL PLBGIN

and no arguments are required.

PLBGIN does not replace PLOTID (see Basic Subroutines), but performs a similar function. PLOTID may be used to write an identification frame on any tape unit, but PLBGIN will only use Tape Unit "A," which is the same tape PLOT uses.

GENERALIZED PLOTTING ROUTINE: PLOT

PLOT performs most plotting tasks, many automatically, and allows a variety of user options that can be expressed in a variable-length calling sequence.

Before PLOT is first used, a tape must be assigned on logical Unit "A," and subroutine PLBGIN must be called to plot an identification frame. After the last frame is plotted, an End-of-File mark must be written on that tape with the FORTRAN statement END FILE 1, or by using the subroutine ENDFLE with an argument of 30.

The user obtains the desired plot by using one of the following calling sequences:

1. PLOT (N,X,Y)

This call will plot N points, each a heavy-mode period, with the coordinates found in the two arrays of floating-point numbers, X and Y.

Scaling will be calculated to fit all the points into the S-C 4020 rectangle bordered by:

(127, 0)	(1023, 0)
(127, 896)	(1023, 896)

A grid of reasonable coarseness (approximately 1/10" between lines) will be overlaid on the graph. Every fifth line will be labeled to fit the computed scaling.

2. PLOT (N, X, Y, OP), where OP is a Hollerith field.

This will perform as 1. above, with the following exceptions:

If OP = 1Hc, the character in the Hollerith field will be plotted where a period was plotted with the simpler call.

If OP = 2HLN, the data will be connected with straight lines.

If OP = 2HQU, the data will be fitted with a single curve by quadratic interpolation.

3. PLOT (N, X, Y, OP, MINX, MAXX, MINY, MAXY)

This call will perform as 2. above, with the exception of automatic scaling.

The S-C 4020 rectangle described under 1. will be scaled by the last four elements of the calling sequence, so that the corner (127, 0) corresponds to (MINX, MAXY).

(127, 896) corresponds to (MINX, MINY)

(1023, 0) corresponds to (MAXX, MAXY)

(1023, 896) corresponds to (MAXX, MINY)

Points not fitting on the total page in this scaling will not appear. If a linear or quadratic curve is called by OP = 2HLN or 2HQU, the curve will go to the edge of the total S-C 4020 area (not merely to the edge of the scaled area) before stopping, and then resume where it would reasonably come back on the page.

4. PLOT (N, X, Y, OP, MINX, MAXX, MINY, MAXY, XINC, YINC)

Besides a planned scale, this call permits the user to determine the coarseness or fineness of his grid.

XINC will be the increment between vertical grid-lines relative to the scaled rectangle; YINC will be the increment between horizontal grid-lines relative to the scaled rectangle.

For legibility, XINC may not be less than (MAXX-MINX)/64; YINC may not be less than (MAXY-MINY)/64.

5. PLOT (N, X, Y, OP, MINX, MAXX, MINY, MAXY, XINC, YINC, NHH, NHV)

This calling sequence performs all operations described under 1-4 inclusive. In addition, two lines of alphanumeric information may be written in the margin of the graph.

NHH should be a normal FORTRAN Hollerith statement, as 11HABCDEFGHIJK, which will print the eleven characters following the H, horizontally 1/2" from the bottom of the page. Subroutine LETTER is used, rather than the S-C 4020 Typewriter.

NHV is the same, but will appear vertically, 1/2" from the left of the page.

In both cases, $N \leq 55$.

If any options are not to be specified, zeros must be entered. CALL PLOT (5, X, Y, 0, 0, 0, 0, 0, . 5., 5) will plot the five X-Y points with gridding 1/2 unit apart, and automatic scaling.

By varying these calls, the user may participate to greater or lesser degrees in functions normally performed with Basic Subroutines. In using the full, twelve-element calling sequence, users will be doing almost (but not quite) as much work as if they employed the Basic Subroutines directly.

Example

The PLOT Subroutine could be used to plot five points at (-1., 2.), (6., 0.), (0., 6.), (3., -1.), and (-5., 4.). The user stores these points in an X array and in a Y array, as follows:

X = (-1., 6., 0., 3., -5)

Y = (2., 0., 6., -1., 4.)

and then calls the Utility Subroutine:

```
CALL PLOT (5,X, Y, 1HX, 0, 0, 0, 0, 0, 0, 15HSPEED AT TIME T)
```

The points will be plotted as X's, because the fourth argument field is 1HX. The grid will be automatically scaled to a square with corners:

(-5., 6.)	(6., 6.)
(-5., -1.)	(6., -1.)

and, since no grid-coarseness was specified, each direction will be divided into sixty-fourths; grid lines will be approximately 1/10 of an inch apart, every .175 horizontal scaled units, and every .1125 vertical scaled units. Every fifth line in each direction will be double-exposed and labeled, i.e., the lines at -5., -4., -3.3, -2.4, etc., will be labeled. In addition, the words, "SPEED AT TIME T," will be written along the bottom of the page. There will be no special writing in the left-hand margin (except the grid-labeling), because the twelfth option was not used.

B. BASIC SUBROUTINES

DEFINING A BUFFER AREA: BUFFER

BUFFER defines a buffer area in core storage, and designates a tape unit, and the camera to be used in handling S-C 4020 plot data. It also allows alternate coordinate scaling for the user not wishing to use S-C 4020 coordinates.

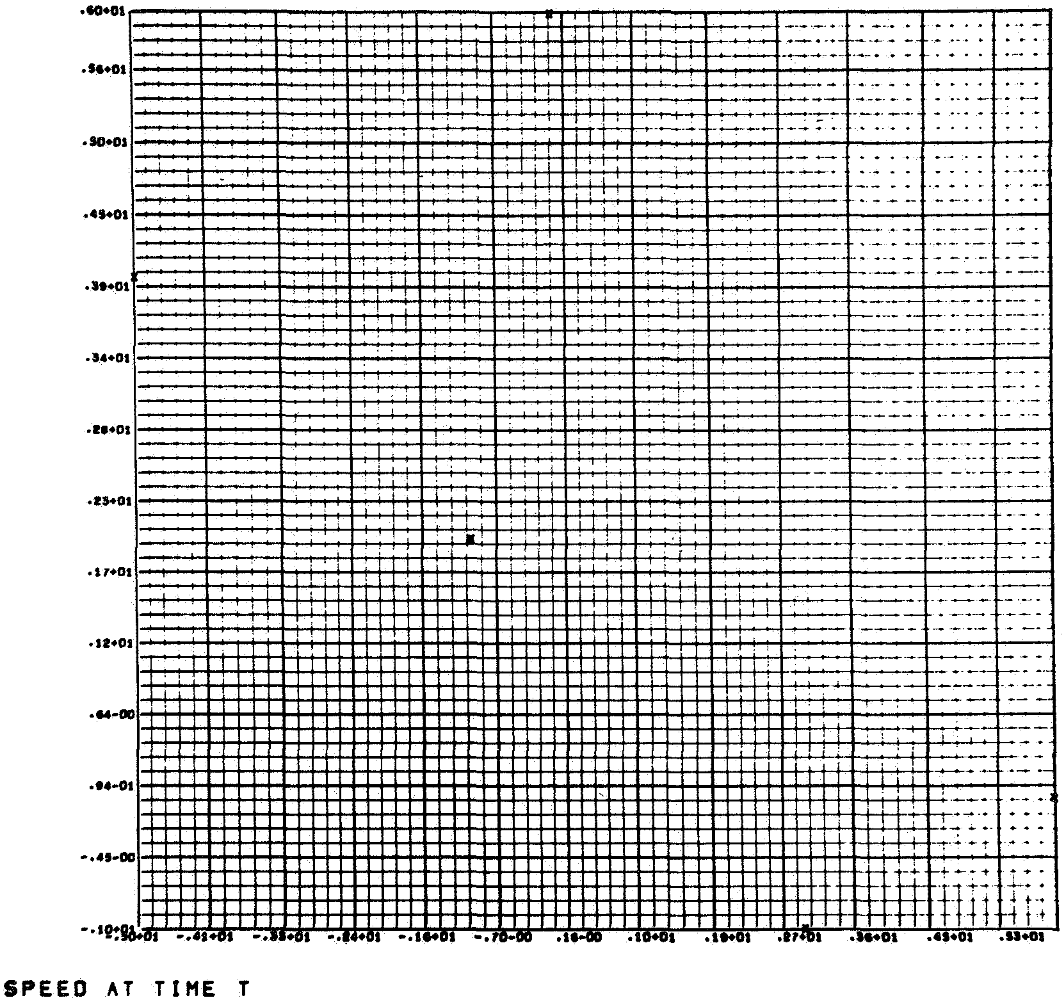


Figure 2-1 Frame Generated by
CALL PLOT (5,X,Y,1HX,0,0,0,0,0,0,15HSPEED AT TIME T)

With regard to restrictions, there are three, namely:

1. The buffer size (ISIZE) must be greater than 10, and not more than 690 (1262_8) words.

2. The tape unit must pick up 120-127 from NTAB$. These numbers correspond to Units A-H.

3. S-C 4020 coordinates are 1024 by 1024, with the origin in the upper left-hand corner; viz.,

$$\begin{array}{ccc} (0, 0) & \rightarrow & (1023, 0) \\ \downarrow & & \downarrow \\ (0, 1023) & \rightarrow & (1023, 1023) \end{array}$$

As for method, the left half of the first buffer location is set to eight less than the size of the buffer. The right half of this location will be used by the Basic Subroutines to store the number of S-C 4020 instructions in the buffer at a given time.

The tape number is stored in the second word.

The third and fourth words are stored with the fixed-point S-C 4020 upper-left X and Y coordinates, respectively.

The fifth and sixth buffer words are stored with the floating-point user's upper-left X and Y coordinates, respectively.

The seventh buffer word is stored with the ratio of the S-C 4020 X coordinates to the user's X coordinates, i.e., (XLRSC-XULSC)/(XLR-XUL).

The eighth buffer word is stored with the ratio of the S-C 4020 Y coordinate to the user's Y coordinate, i.e., (YLRSC-YULSC)/(YLR-YUL).

The subroutine's calling sequence is

CALL BUFFER (BUF, ISIZE, ITAPE, ICAM, XULSC, YULSC, XLRSC, YLRSC, XUL, YUL, XLR, YLR)

where:

BUF	an available array larger than eight words
ISIZE	number of cells in the BUF array $\leq$ 690 (1262_8)
ITAPE	tape number
ICAM	camera number (1, w, or 3). Number 3 means both 1 & 2
XULSC	upper-left S-C 4020 X coordinate (fixed or floating)
YULSC	upper-left S-C 4020 Y coordinate (fixed or floating)
XLRSC	lower-right S-C 4020 X coordinate (fixed or floating)
YLRSC	lower-right S-C 4020 Y coordinate (fixed or floating)
XUL	upper-left user's X coordinate (floating)
YUL	upper-left user's Y coordinate (floating)
XLR	lower-right user's X coordinate (floating)
YLR	lower-right user's Y coordinate (floating)
NOTE:	If all coordinates will be entered in the S-C 4020 coordinate system, the last eight elements of the calling sequence may be deleted.

This subroutine must be used to define a buffer area before using any of the S-C 4020 Basic Subroutines that require such a buffer. Any number of these buffers may be defined, but when a tape number is duplicated in more than one buffer, special care must be used to ensure that plot instructions are in proper order. Proper use of the subroutine RITE should be useful for this purpose.

Example

CALL BUFFER (BUF, 690, 30, 1, 0, 0, 1023, 1023, -1., 1., 1., -1.) will set up a buffer region at location BUF (which must be previously dimensioned 690), in which all user-scaled values will be adjusted as if the plotter frame,

(0, 0)	(1023, 0)
(0, 1023)	(1023, 1023)

were:

(-1, 1)	(1, 1)
(-1, -1)	(1, -1)

which is a much more likely user-scheme of grid-orientation.

WRITING AN IDENTIFICATION FRAME: PLOTID

PLOTID writes an identification frame on the tape to be plotted, marking each run with the user's identification and charge number (taken from the RUN card), and the date and time. PLOTID should be called once at the beginning of each distinct run.

A buffer must previously have been defined by subroutine BUFFER.

PLOTID uses an entire frame, advancing the file before returning to the master program.

The 1107/1108 Monitor picks up information from the RUN card and the real-time clock; this information is plotted with subroutine LETTER.

The calling sequence is

```
CALL PLOTID (BUF)
```

where BUF is a buffer region previously defined by subroutine BUFFER.

PLOTID, which generates a title page for each run, should be the first subroutine called after BUFFER. PLOTID advances the film one frame, so that user data begins on the second frame.

Example

If Subroutine BUFFER had been called to define a buffer region at core location BUF,

CALL PLOTID (BUF)

would generate an identification frame. For a user with RUN card information reading

ΔbRUNb5USER, 022704, 5, 25

the identification frame would read

5USER		
022704	TIME	DATE

NOTE: This routine will change as the system run cards are changed.

GENERATING A GRID: GRID

This routine prepares instructions for the S-C 4020 to produce an M by N grid. Every MAJX horizontal line and every MAJY vertical line will be heavily exposed.

The second calling sequence for GRID shows two optional arguments, NXU and NYU, which may be coded to act as stop points for the X and Y axes. If these arguments are not coded, each axis will sweep out to the edge of the frame.

The buffer region must have been assigned previously. The coordinates of the lower-left corner of the grid must be in fixed-point arithmetic and are assumed to be in the S-C 4020 coordinate system.

The calling sequence for GRID is

CALL GRID (BUF, NX, NY, MAJX, M, MAJY, N)

or

CALL GRID (BUF, NX, NY, MAJX, M, MAJY, N, NXU, NYU)

where:

BUF	address of an array previously defined by subroutine BUFFER
NX	address of lower-left corner X coordinate
NY	address of lower-left corner Y coordinate
MAJX and MAJY as described above	
M	address of number of divisions to take along the X axis
N	address of number of divisions to take along the Y axis
NXU	address of upper-right corner X coordinate
NYU	address of upper-right corner Y coordinate

DRAWING ALPHANUMERIC AND SPECIAL CHARACTERS: LETTER, LETTR2

These subroutines prepare instructions for the S-C 4020 to draw alphanumeric and special characters. The difference between these and other print routines is that the characters produced by LETTER and LETTR2 may differ in height and width. Subroutine LETTR2 is a version of LETTER that draws letters, numbers, and characters sideways.

The only restriction is that the buffer region must have been assigned previously.

The height of any character is 12* MAGNY, and the width is 8* MAGNX S-C 4020 units. The spacing between letters written horizontally will be 8* MAGNX S-C 4020 units, and vertically will be 4* MAGNY S-C 4020 units. Only one line of characters will be constructed each time LETTER is called, but this line of characters may run vertically or horizontally. The user must provide the coordinates (X, Y) of the lower left corner of the first character of a line. Quantities X, Y, MAGNX, and MAGNY may be in either fixed- or floating-point arithmetic. However, all quantities given in fixed-point arithmetic are assumed to be in the S-C 4020 system.

Illegal characters will be replaced by blanks.

The calling sequence for LETTER is

CALL LETTER (BUF, X, Y, MAGNX, MAGNY, SCALE, N H CHARACTERS)

where:

BUF address of an array previously defined by BUFFER

(X, Y), MAGNX and MAGNY as defined above

MAGNX<0 indicates the characters are to follow in a vertical fashion

SCALE address of scaling indicator

If floating-point arithmetic is used, SCALE = 0 indicates the user is using the S-C 4020 coordinate system.

N H CHARACTERS Gives the address of the array of Hollerith Characters to be plotted. The last word of the array will be filled with sevens.

while that for LETTR2 is

CALL LETTR2 (BUF, X, Y, MAGNX, MAGNY, SCALE, N H CHARACTERS)

where:

The arguments represent the same items as for LETTER, except that the (X, Y) coordinates determine the upper left-hand corner of the first symbol, and all output will be rotated ninety degrees counterclockwise.

PLOTTING A CHARACTER: PLOTP, PLOTH, PLOTL AND PLOTI

These routines will plot a BCD character at a specified point in either a light or heavy exposure, or in an exposure of a specified intensity.

The only restriction is that the buffer region must have been assigned previously by subroutine BUFFER.

These routines are entry points in subroutine BOOK, which is used to store S-C 4020 instructions in the buffer region.

PLOTH will plot the BCD character in heavy exposure, PLOTL in light exposure, and PLOTP in the existing exposure level. PLOTI is a newly added routine for plotting a BCD character in a specified intensity within the range of 0 (lightest) to 15 (darkest). The calling sequence for PLOTI is the same as for PLOTL, PLOTH, and PLOTP with the exception of an additional parameter, INT, which specifies the address of the intensity array.

The user may specify a single point or an array of points at which the character indicated by 1HC is to be plotted. All S-C 4020 characters in figure 8 are available in the plotting mode.

The coordinates may be in the user's or the S-C 4020 coordinate system. All coordinates not given in the S-C coordinate system must be in floating-point arithmetic. Otherwise, either fixed- or floating-point arithmetic may be used.

The calling sequences are

```
CALL PLOTH ⎫
CALL PLOTL ⎬   (BUF, N, X, Y, DELX, DELY, 1HC, ISCALE)
CALL PLOTP ⎭

CALL PLOTI     (BUF, N, X, Y, DELX, DELY, 1HC, ISCALE, INT)
```

where:

BUF	address specified by BUFFER
N	number of points at which the character is to be plotted
X	address of the X array
Y	address of the Y array
DELX	value to be added to each X each time to generate an X array

DELY	similar to DELX but for Y
1HC	C is the character to be plotted at each (X, Y)
ISCALE	0 if the S-C coordinate system is used
INT	the address of the intensity array

Example

To plot five crosses, equally spaced at (1, 2, 3, 4, 5) horizontally, and positioned at $y = \sqrt{X}$ vertically, compute and store the vector ($\sqrt{1}$, $\sqrt{2}$, $\sqrt{3}$, $\sqrt{4}$, $\sqrt{5}$) in Y. Assuming an appropriate scaling (from an earlier call of BUFFER),

```
CALL PLOTP (BUF, 5, 1., Y, 1., 0., 1HX, 1)
```

CONNECTING POINTS WITH LINEAR VECTORS: PLOTVL

This subroutine connects the data points with linear vectors and stores the generated plot instructions into a previously defined buffer region.

Regarding PLOTVL, there are three restrictions:

1. The curve number must be in the range 1-30.

2. Points with the X coordinate out-of-range will be ignored.

3. Points with the Y coordinate out-of-range will produce a vector to the border at an interpolated point. Another point will be computed on the border by use of the next point with valid Y coordinate. The graph will then resume from this point.

As for method, to allow the subroutine to be used to form graphs on more than one tape simultaneously, and to enable the user to enter a minimum of one point and yet obtain a graph composed of linear segments, a curve number is required for graph identification. This number is assigned by the user and may be re-assigned as soon as a graph is completed. If the graph number is used, the last point of a data group is saved and identified by its curve number to permit more points to be connected by a later reference to the subroutine. Up to 30 graphs may be built simultaneously, and the curve number may be reused once a graph is completed.

PLOTVL's calling sequence is

CALL PLOTVL (BUF, NUM, XADD, YADD, DELX, DELY, IST, ISCALE, ICURVE)

where:

BUF	an array previously defined in subroutine BUFFER
NUM	number of points
XADD	address of first X coordinate
YADD	address of first Y coordinate
DELX	if non-zero, add to (XADD) to get successive X coordinates if zero, assume the X coordinates are in an array beginning at (XADD)
DELY	if non-zero, add to (YADD) to get successive Y coordinates if zero, assume that Y coordinates are in an array beginning at (YADD)
IST	1, join to the previous point on curve ICURVE 0, this call states a curve
ISCALE	if non-zero, coordinates will be scaled according to the call of subroutine BUFFER for BUF. Coordinates must be floating-point. if zero, S-C 4020 coordinates are assumed. Coordinates may be fixed- or floating-point
ICURVE	curve number used for linking with previous point. May be reassigned upon completion of a curve. If IST = 0, this number is assigned to the curve. Use 1, 2, . . ., 30.

In drawing vertical or horizontal lines, the call

CALL PLOTVL (BUF, 2, XADD, YADD, 0, DELY, L, K, J)

will imply that two identical points are stored in XADD, XADD+1 (or YADD, YADD+1 if DELY = 0. for horizontal lines); i.e., a DELX or DELY of zero violates the calling sequence restrictions, or, more to the point, the solution is set

$$X(1) = X(2)$$

and call

CALL PLOTVL (BUF, 2, X, YADD, 0, DELY, L, K, J)

Similarly for horizontal lines (DELY = 0.).

GENERATING A QUADRATIC CURVE: PLOTVQ

To generate a set of instructions for the S-C 4020, which will approximate a quadratic through each successive triple of given data points.

For PLOTVQ, there are four restrictions,

1. The curve number must be in the range $1 \leq \text{ICURVE} \leq 8$.

2. The data points must be in floating-point if scaling is indicated.

3. The buffer area used must have been previously defined by the BUFFER subroutine.

4. No instructions are generated until the third point of a curve has been received by the subroutine.

Four considerations regarding method should be stated.

1. The points for a specific curve are stored until three points are available.

2. Additional points are generated using quadratic interpolation.

3. As each new data point is added, the additional points are added using quadratic interpolation based on the new point and the two previous points.

4. The curve number is used for building several curves simultaneously.

The calling sequence for PLOTVQ is

CALL PLOTVQ (BUF, NUM, XADD, YADD, DELX, DELY, IST, ISCALE, ICURVE)

where:

BUF	an array previously defined by subroutine BUFFER
NUM	number of points
XADD	address of X coordinate of first point
YADD	address of Y coordinate of first point
DELX	if non-zero, add to (XADD) to get successive X coordinates if zero, assume the X coordinates are in an array beginning at (XADD)
DELY	if non-zero, add to (YADD) to get successive Y coordinates if zero, assume that Y coordinates are in an array beginning at (YADD)
IST	0 means first point is first point of curve 1 means first point is second point of curve 2 for all other cases
ISCALE	0 means coordinates are in S-C 4020 system 1 means all coordinates will be scaled
ICURVE	curve identification number

The following items may help to use the subroutine properly.

1. If the coordinates are not to be scaled, they may be either fixed- or floating-point. Numbers to be scaled, however, must be floating-point.

2. Two curves may be assigned the same curve number if all points are given for one, before any points are given for the second.

3. If points are being given for two curves simultaneously, two distinct buffer and tape designations are required.

Example

The example under PLOTP will illustrate PLOTVQ. If X = (1, 2, 3, 4, 5) and Y = ($\sqrt{1}$, $\sqrt{2}$, $\sqrt{3}$, $\sqrt{4}$, $\sqrt{5}$), the square root curve may be plotted by

CALL PLOTVQ (BUF, 5, 1., Y, 1., 0., 0, 1, 1)

The fifth argument (1.) will be added to the initial X value for successive points, while the sixth argument (0.) requires the ordinates to be taken from the Y array.

SETTING COUNT VALUE: SET

To set COUNT for use with PTYPE and statements of the form:

WRITE (Tape, Format) List

and to allow the use of subroutine LETTER under the control of such WRITE statements.

The COUNT-value must be a valid Y-coordinate in the S-C 4020 system, $0 \leq Y \leq 1023$, fixed- or floating-point.

If the long calling sequence is used, the instructions under LETTER must be followed. In particular, this long call must be preceded by a call of BUFFER.

The calling sequence to set COUNT is

CALL SET (Y)

where:

Y = S-C 4020 coordinate to be set into COUNT.

The long call is

CALL SET (BUF, X, Y, MAGNX, MAGNY, ISCALE)

where all six arguments are identical to the first six elements in a call of LETTER.

Call SET with one argument just before calling PTYPE or WRITE, to position the next output on the S-C 4020 grid.

If the S-C 4020 Typewriter is used, Figure 2-2 should be consulted regarding differences between 1107/1108 and Typewriter characters.

Use the long call of SET just before a WRITE statement to execute that WRITE statement through LETTER, rather than through the S-C 4020 Typewriter.

TYPING INFORMATION ON GRID: PTYPE

This routine will type information on the S-C 4020 grid at a specified location.

A buffer area must have been assigned previously by subroutine BUFFER.

If the short calling sequence is used, the starting X-coordinate is set to zero. The Y-coordinate is determined by:

1. Adding 16 S-C 4020 coordinate units to the value previously entered by the subroutine SET.

2. Adding 16 S-C 4020 coordinate units to the previous position of a WRITE (I, F) LIST, whichever was most recent.

The calling sequence is

CALL PTYPE (BUF, ARRAY, INUM, XADD, YADD, ISCALE)

or

CALL PTYPE (BUF, ARRAY, INUM)

where

BUF	a buffer area previously defined by subroutine BUFFER
ARRAY	an array containing the alphabetic information to be typed in Field Data code
INUM	number of words in ARRAY
XADD	X-coordinate to start typing
YADD	Y-coordinate to start typing
ISCALE	{ 0 indicates no scaling non-zero indicates scaling

The last three terms of the calling sequence may be omitted as described under the requirements mentioned previously.

The Field Data Code representations of characters will be translated to Binary Coded Decimal and rendered according to the following table of S-C 4020 Typewriter characters. The user should note the differences between S-C 4020 and UNIVAC 1107/1108 fonts.

ADVANCING THE FILM: VANCE

This routine will advance the film one frame in the selected camera(s).

The programmer should only remember that the buffer array must have been assigned previously.

Field Data Code	Binary Coded Decimal	Card Code	S-C 4020 Character	UNIVAC 1107/1108 Character
00	52	7-8	. (*)	@
01	56	12-5-8	~ (*)	[
02	72	11-5-8	o	]
03	12	12-7-8	∂ (*)	#
04	57	11-7-8	d	Δ
05	60	Blank	Blank	Blank
06	21	12-1	A	A
07	22	12-2	B	B
10	23	12-3	C	C
11	24	12-4	D	D
12	25	12-5	E	E
13	26	12-6	F	F
14	27	12-7	G	G
15	30	12-8	H	H
16	31	12-9	I	I
17	41	11-1	J	J
20	42	11-2	K	K
21	43	11-3	L	L
22	44	11-4	M	M
23	45	11-5	N	N
24	46	11-6	O	O
25	47	11-7	P	P
26	50	11-8	Q	Q
27	51	11-9	R	R
30	62	0-2	S	S
31	63	0-3	T	T
32	64	0-4	U	U
33	65	0-5	V	V
34	66	0-6	W	W
35	67	0-7	X	X
36	70	0-8	Y	Y
37	71	0-9	Z	Z

NOTE: (*) Not available in typewriter mode.

S-C 4020 Characters Compared with 1107/1108 characters

Figure 3-2 (Part 1 of 2)

Field Data Code	Binary Coded Decimal	Card Code	S-C 4020 Character	UNIVAC 1107/1108 Character
40	34	12-4-8	)	)
41	40	11	-	-
42	20	12	+	+
43	17	12-6-8	α	<
44	13	3-8	=	=
45	35	6-8	β	>
46	76	2-8	Σ	&
47	53	11-3-8	$	$
50	54	11-4-8	*	*
51	74	0-4-8	(	(
52	14	0-5-8	"	%
53	55	5-8	γ	:
54	37	12-0	?	?
55	16	11-0	δ	!
56	73	0-3-8	,	,
57	36	0-6-8	±	\
60	00	0	0	0
61	01	1	1	1
62	02	2	2	2
63	03	3	3	3
64	04	4	4	4
65	05	5	5	5
66	06	6	6	6
67	07	7	7	7
70	10	8	8	8
71	11	9	9	9
72	15	4-8	,	,
73	32	11-6-8	π	;
74	61	0-1	/	/
75	33	12-3-8	°	.
76	77	0-7-8	⌑	⌑
77	75	0-2-8	∫	Idle

Figure 3-2 (Part 2 of 2)

This routine is an entry point in subroutine BOOK. The latter is used to store the instruction in the buffer.

The calling sequence for VANCE is

CALL VANCE (BUF)

where

BUF an array previously defined by subroutine BUFFER.

WRITING PLOT INSTRUCTION TO TAPE: RITE

This routine will empty onto tape all plot instructions stored in the specified buffer.

The buffer region must have been previously defined by subroutine BUFFER.

The tape number and plot instruction count is picked up from the buffer region. The subroutine TWR$ is used to write the instruction onto tape.

The subroutine TCHK$ is used to verify a normal write. A return to MERR$ is made when an error or abnormal exit occurs.

The calling sequence is

CALL RITE (BUF)

where:

BUF a buffer region defined by subroutine BUFFER

This subroutine may be used at any time to ensure that the plot instructions are written on tape in a particular sequence. For example, a "WRITE (Tape, Format) LIST" writes plot instructions directly on tape without using a buffer area defined by Subroutine BUFFER; such plot instructions would then precede any instructions still in the buffer region. A "CALL RITE (BUF)" just before the write statement would preserve proper order.

RITE must also be used following the last reference to plot subroutines to ensure that all plot instructions are written on tape.

If BUF contains no plot instructions when RITE is called, return is made immediately.

PROJECTING A FORM SLIDE: JECT

When Form Slides are used to superimpose a picture on the S-C 4020 output, Subroutine JECT is called to project the Form Slide and the data (from normal plotting) on the film. For information about Form Slides, see the section entitled, GENERAL OPERATING PROCEDURES, page III-5.

The buffer region must have been assigned previously.

This routine is an entry point in subroutine BOOK, which is used to store the instruction in the buffer.

The calling sequence for JECT is

CALL JECT (BUF)

where:

BUF	an array previously defined by subroutine BUFFER.

SELECTING A CAMERA: CAMERA

This routine will open the shutter on the selected camera and close the shutter on the remaining camera. However, the buffer area must have been assigned previously.

This routine is an entry point in subroutine BOOK.

The calling sequence for CAMERA is

CALL CAMERA (BUF, N)

where:

BUF	buffer address
N=1	(open shutter on F80 camera)
N=2	(open shutter on 35mm camera)
N=3	(open shutter on both cameras)

Camera selection is an operator function and is determined by the user on the Plot Request Form. If Subroutine CAMERA is called, the user must request the operator to transfer the Plotter to Programmer Control.

CHANGING THE IMAGE SIZE: EXPAND AND REDUCE

EXPAND and REDUCE are two recently added subroutines for changing the image size on the CHARACTRON tube face. EXPAND increases the image size from normal to provide frame abutment (for strip charting), while REDUCE decreases the image size to normal.

The calling sequences for these two routines are simply

CALL EXPAND (BUF)
CALL REDUCE (BUF)

where:

BUF address specified by BUFFER

WRITING AN END OF FILE TO PLOT TAPE: ENDFLE

This subroutine writes an end-of-file on a plot tape.

The logical tape number must be a position in NTAB$ that corresponds to a value in the range 120-127. These in turn correspond to Tape Units A-H.

The calling sequence is

CALL ENDFLE (N)

where N contains a logical plot tape number.

An end-of-file must terminate each run. However, this subroutine in no way interferes with the usual end-of-file statements.

SECTION IV

OPERATIONAL PROCEDURES

OPERATIONAL PROCEDURES

UNIVAC 1107/1108 EXECUTIVE SYSTEM

Both the Lockheed and the CSC packages were implemented for use on the 1107/1108 which operates under the control of a resident monitoring system called EXEC II. Both packages are FORTRAN IV compatible. The Lockheed version is contained on the system library while the CSC version is contained on the auxiliary library tape.

The EXEC II monitor for the 1107 is described in GSFC document number X-543-64-212. The EXEC II monitor for the 1108 is explained in Univac's manual number UP-4058.

IBM 7090/7094 EXECUTIVE SYSTEMS

As stated previously, two packages exist for the IBM 7090/7094 computer: SCORS II, which is FORTRAN II oriented, and SCORS IV, oriented toward FORTRAN IV. SCORS II and SCORS IV are on both the FMS and IBLIB system libraries respectively.

SCORS II may be called upon via the FMS Executive (FORTRAN Monitoring System), while SCORS IV operates under the FORTRAN IV Executive called IBSYS. Neither FMS nor IBSYS are explained within this document.

SAMPLE COMPILATION AND RUN DECKS

The following setups illustrate deck structures in the simplest forms. The cards are not explained and all the required cards for a particular job are not indicated. Available documentation for the various systems should be consulted.

A. UNIVAC 1107/1108 EXEC II DECK

This simplified deck structure illustrates a simple compilation with card input, no test data and output assigned to tape unit H.

```
∇  RUN identification  account
∇  ASG  H = output
∇  FOR  n1/v1
     CALL IDFRMV etc.
     CALL CAMRAV etc.
            .
            .                      Main program to be
     main program, including      compiled/assembled
     the termination call to
     EOFTV
     CALL EOFTV
            .
            .
∇ XQT  n1
∇ FIN
```

Note: ∇ is a seven with an eight overpunch.

B. IBM 7094 FMS DECK (SCORS II, FORTRAN II)

This sample again illustrates a simple compilation. The executive monitor is FMS and the object program's output unit is A7.

```
Columns  1    2    7
         *         DATE mm/dd/yy
         *    identification incl. name and phone number
         *         PAUSE
         *         XEQ
```

```
*            CARDS COLUMN (required if binary deck is desired)
*            LIST8
*            FORMAP
                 .
                 .
                 .
             main program including
             termination call to PLTND
                 .
                 .
                 .
∇
```

C. IBM 7094 IBSYS DECK (SCORS IV, FORTRAN IV)

Another sample compilation, but it is to be run under the control of the IBSYS monitor. Output through the SCORS IV package is directed to tape unit A7.

```
Columns  1         7         16
         $IBSYS              (may be omitted)
         $JOB      identification information
         $PAUSE
         $EXECUTE            IBJOB
         $IBJOB              MAP, GO
         $IBFTC    job name  LIST, REF
           .
           .
           .
           main program including
           termination call to PLTND
           .
           .
           .
         ∇
```

REQUIRED FORMS

In order to run a job on one of the computers to produce a plot tape for the S-C

4020 Recorder, certain forms, available at the GSFC Dispatcher's desks must be completed by the programmer.

A. INSTRUCTION CARD (1107/1108 and 7094)

Two forms of instruction cards exist - one form for the 1107/1108 and the other for the 7094. One form, depending upon which computer is to be employed, must be completed and submitted with the programmer's package. Sample cards are shown in Figures 4-1, 4-2 and 4-3.

B. S-C 4020 REQUEST FORM

After a plot tape has been generated, it must then be removed for processing on the S-C 4020 Plotter and a request form exists for this procedure. A self-explained sample appears in Figure 4-4. It is recommended that this form be filled out after the plot tape has been generated and submitted separately, rather than to expect this form to be filled out and forwarded by computer operations personnel. Hence, the programmer should indicate on his Instruction Card that his plot tape is to be retained on file and is not to be "scratched" after his run. The programmer's full name, not just initials must appear on the S-C 4020 Request Form.

PROGRAMMER ID PROGRAMMERS NAME PHONE________ PROGRAMMER PRESENT ☐

7094 MODE ☐ 1 ☐ 2 TIME___HR___ ___MIN

LOGICAL	A1		A2	A3		A5	A7		B2					
TAPE NUMBER	IBSYS		INPUT				PLOT							
DISPOSITION	LFR	LFR	LFR	LFR	LFR	LFR	LFR	LFR	LFR	LFR	LFR	LFR	LFR	LFR

KEYS ☐☐☐☐☐☐☐☐☐☐☐☐ SENSE ☐☐ ☐ 65K ☐ DISK

CHANNELS ☑ A ☑ B ☐ C ☐ DCC ☐ CARD READER ☐ PRINTER ☐ PUNCH

EXPECTED STOP ☐ DUMP IF MAX. TIME ☐ TROUBLE

SPECIAL INSTRUCTIONS:

DELIVER A7 TO SC4020 PLOTTER WITH ATTACHED REQUEST FORM

7094 INSTRUCTION CARD

TO DISPATCHER: BLDG 3 ROOM 159

PERIPHERAL I/O INSTRUCTIONS

☑ STANDARD CARD TO TAPE

PRINT OUTPUT TAPES

LOGICAL	1 2 3 PC	FILES	COPIES	FORM	CARRIAGE LOOP
A3	PC	1	1	W	

PUNCH TAPES

LOGICAL________ FILES______ CARD FORM #______

LOGICAL________ FILES______ CARD FORM #______

GSFC 6-54 (6 '66)

RETURN TO: BLDG_____ ROOM_____

A N N N N N

SPONSOR NUMBER

A A A A A

PROGRAMMER ID

N N N N N

PROJECT NUMBER

A CATEGORY A TYPE

PROGRAM NUMBER

☑ 65K

Instruction Card-IBM 7094

Figure 4-1

1107/1108 INSTRUCTION CARD

TASK ______ OF RUN ______

SPONSOR NO.	PROGRAMMER ID	PROJECT NO.	CAT	TYPE	PROGRAM NO.	NAME	PHONE
ANNNN	AAAAA	NNNNN	A	A		FULL NAME	

SCHEDULING: EST. TIME (MIN.) 15 EST. PAGES ______ ☐ ATTENDED RUN

COMPUTER: ☑ 1107 ONLY ☐ 1108 ONLY ☐ 1107/1108

MONITOR: ☑ EXEC II ☐ 1108 EXEC OTHER ______

NOTES:

I/O OUTPUT: ☐ CARDS ☑ TAPES ☐ PRINTER OTHER ______

I/O INPUT: ☑ CARDS ☐ TAPES OTHER ______

SETTINGS: JUMP KEYS (circle) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

HALT KEYS (circle) 1 2 3 4

OFFICE LOG NUMBER

GSFC 6-49 (6/66)

Execution Instructions (Production or Utility Program Name ______)

Normal Stops: ☑ Endrun ☐ Stop Other

Error Procedures (If max time, max pages, etc.)

Instruction Card - Univac 1107/1108

Figure 4-2

1107/1108 TAPE SETUP CARD NAME FULL NAME

Operator Setup (pencil)		Programmers Logical Setup		Optional Specifications					
CHAN	UNIT	ASG LABEL	REELS	SAVE OUTPUT	LOGICAL CHANNEL	ERROR ACTION*	ENABLE WRITE	DENSITY	
		H=PLOT	(BLANK)	✓			✓	HI	

*Standard Error Action (leave column blank):
Attempt recovery, finally fault record

Non-Standard Error Action
1. ______
2. ______

GSFC 6 50 16 661

Operator Setup (pencil)		Programmers Logical Setup		Optional Specifications					
CHAN	UNIT	ASG LABEL	REELS	SAVE OUTPUT	LOGICAL CHANNEL	ERROR ACTION*	ENABLE WRITE	DENSITY	

Tape Setup Card-Univac 1107/1108

Figure 4-3

GODDARD SPACE FLIGHT CENTER
DATA PROCESSING BRANCH – CODE 545

REQUEST FOR SC-4020 OPERATION

NAME	PHONE	DATE
Programmers Full Name	PHONE #	DATE
CODE	SATELLITE	
CODE NO.	AS REQUIRED	

INPUT

HI DENSITY	☑
LOW DENSITY	☐
BINARY	☑
BCD	☐
PRINT NORMAL	☐
PRINT LIST	☐

TAPE # 4020 PLOT TAPE #

NO. FILES # NO. FRAMES #

OUTPUT

HARDCOPY		☑
FILM	AS YOU SELECT	☐
NEGATIVE		☐
POSITIVE		☐

DUPLICATES

HARDCOPY	☐
FILM	☐

Special Instructions ____________________

Deposition of Output ____________________

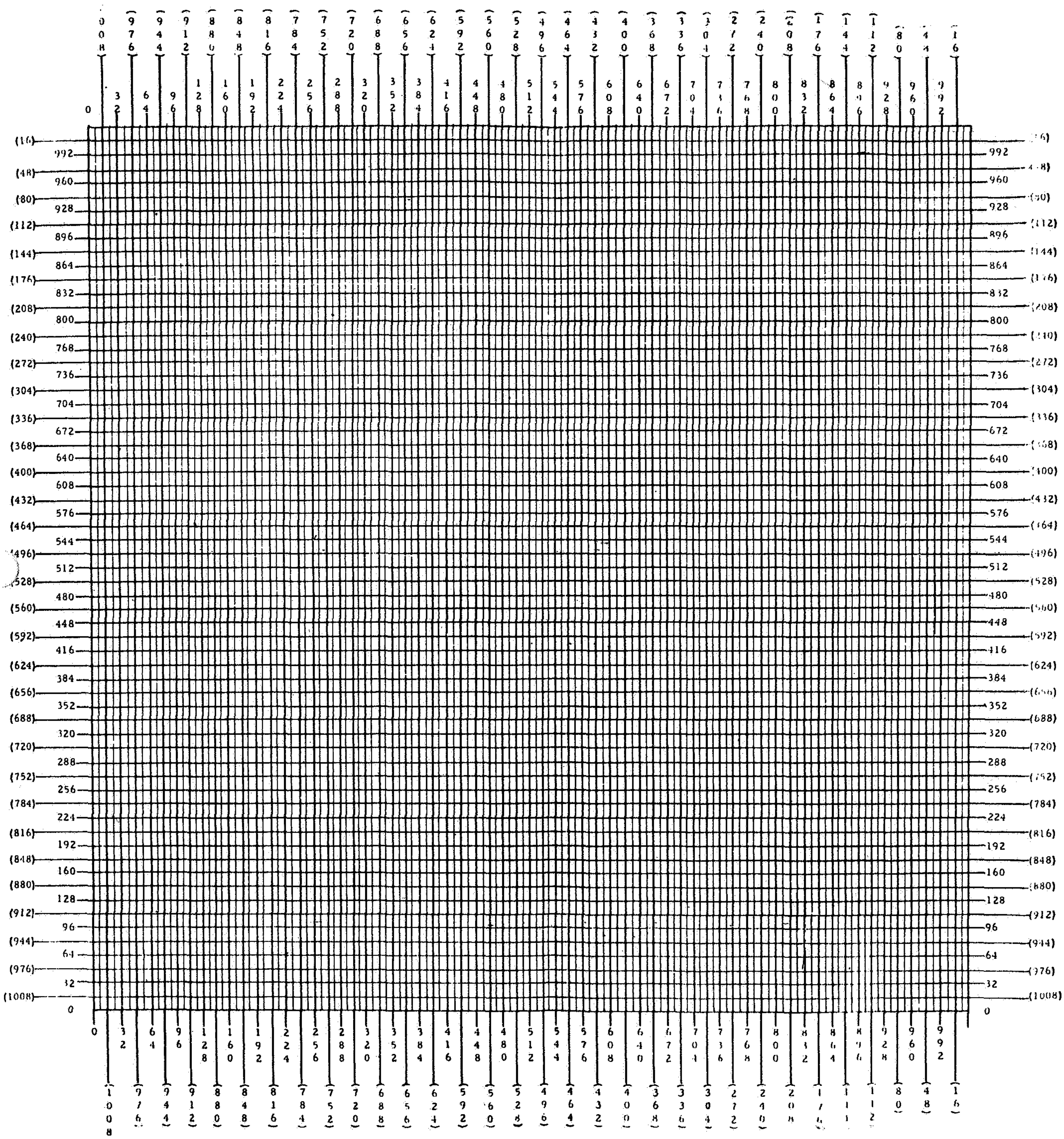
Operator Remarks ____ Name ____________________ Date Completed

S-C 4020 Request Form

Figure 4-4

540-67 (2/65)

This is an example of an S-C 4020 Plot Layout Worksheet.



OUTPUT CONSIDERATIONS

In order to better understand the possible outputs available for the SC-4020, some pertinent characteristics should be known.

First, it can produce film, paper, or film and paper during one pass of the input magnetic tape.

Second, the 4020 was designed to produce a film output and as such is not the best paper producer. The consistency of paper output cannot be held within close tolerance, for only the film output can be checked for intensity variation across the intensity band of the machine. This check is accomplished a number of times a day (twice/shift) along with checks on all other succeeding steps. It should therefore be noted that this machine may not produce a paper output which results in a consistent copy on any other media. The film will, however, result in a good hard copy.

Third, in order to insure the highest quality output for all users, no modifications to "F Stop" or "Tube Intensity" will be made. These have been set to the optimum values for this machine. Any variation from normal operation must be done under program control. Any variation from normal procedures must be in writing and approved by the Section or Branch Head. This work will be done on the swing shift (1600 to 2400).

Fourth, the rotation of the tube face 90^{o} for motion pictures will be done by the operator on receipt of a written request. No prior authorization is necessary.

The following turn around times shall be adhered to for normal testing and programmer debugging during the prime shift (0800 to 1600).

Type of Work*	Time Submitted	Work Completed
Negative Film	Before 1100	1330
Positive Film**	Before 1100	24 hrs.
Paper	Before 1100	1330
Negative Film	Between 1100 and 1500	1700
Positive Film**	Between 1100 and 1500	48 hrs.
Paper	Between 1100 and 1500	1700

Under normal circumstances jobs will be run as they are received; however, special priority work can be handled. These special priority jobs are those requiring rapid turn around or those submitted after normal working hours. In both of these cases a normal request should be filled out as outlined in the referenced page IV-4, (submitted during normal working hours) and the priority noted in writing on the request by either the senior Production Controller (F. Greer) or the Section Head (M. Selig). If neither is available, the request can be signed by any other member of the Production Control Group.

* These jobs are non-production and to insure all users of the 4020 equal opportunity to obtain their output, these jobs should be as short as possible. Any job which will exceed the capacity of one roll of film or paper (4000 frames of film or 300 frames of paper) shall be considered a production job and run during the swing shift (1600 to 2400).

** These jobs are processed by an out-of-house contractor and daily pick up and delivery is at approximately 1200 and the finished product is returned the next day at about 1200.

The minimum time to process either paper or film is twenty minutes plus the time to run the 4020. The 4020 set-up time is five (5) minutes and the processing rate is 8-10 frames per second for film and one frame per three (3) seconds for paper. If paper and film output together, the rate is the same as that for paper.

At this time we do not have the capability to produce duplicate negatives from a master (this capability is being acquired). We can, however, produce a positive copy.

All jobs should be programmed in such a way that a title block is generated which contains the programmer's name, the project, and any other pertinent information. This will clearly indicate the start of each job and will ensure your receipt of the entire job. It is also important that all long runs have instructions attached which indicate a point at which the job can be cut if necessary. These jobs will be spliced together again. All jobs in excess of 1000 frames (film) should have a blank in frame 1001 and at every 1000 frames. This will allow the film to be placed on the 100 foot reels.

SECTION V

OTHER PROGRAMMABLE OPERATIONS

OTHER PROGRAMMABLE OPERATIONS

A. RETENTION OF PLOT COMMANDS

Occasionally, some complicated programming effort is required to produce a particular plot image which requires continual use as a background. In such a case, it is far more practical to retain the results of the computations than to recompute the required input data necessary to construct the particular background image upon which new data points will be plotted. Plot commands which generate a particular background are saved on an intermediary buffer/work tape or drum from whence the commands are retrieved.

A series of routines for the 1107/1108 are in existence for the storage and retrieval of plot commands. They are ONSAV, OFFSAV and GETSAV. All are explained in Appendix B of this document. As explained, the storage is accomplished via work tape or drum, which is preferred.

B. MULTIPLE PLOTS/FRAME

This is to recapitulate the earlier comments concerning GRID1V, SETMIV and SETMOV of the SCORS and Lockheed packages.

Occasionally, it is more practical to generate more than one plot per frame. Since GRID1V normally reserves a strip, 24 raster counts in width, at the top, bottom and left of each grid for title displays, a means was necessary to provide marginal variation. SETMIV and SETMOV provide such a means and are explained earlier in Section II, page II-25. With this capability, it is quite possible to have more than a single plot per frame.

With the CSC package, for the UNIVAC 1107/1108 multiple plots can be accomplished by using the basic subroutine BUFFER, to set the graph boundaries, and the various other basic subroutines.

C. STEREOGRAPHIC PROJECTION [1]

The Weatstone stereoscope consists of a pair of mirrors each of which is so oriented as to reflect a different view of a common scene. The Brewster stereoscope consists of a pair of lenses, each of which is focused on a different view. The left eye differs from the right eye view by such an amount that the two views appear to merge into a single image with depth. The two views constitute a stereogram.

The perception of depth depends upon the differences between angles which are subtended by details in the two views of the stereogram. The limit of binocular perception of depth is 30" of arc[2]. If Traid Camera film is viewed in a conventional stereoscope, the angular distance between dots is 74" of arc. Insofar as the plotted views are nearly as fine as the limit of perception, the S-C 4020 Recorder is capable of making good stereograms.

However, a routine is required which plots datum points on the S-C 4020 Recorder to form each of the two views of a stereogram. The following formulation for such a routine was taken from NWL Report No. 1686 (mentioned previously).

Let x, y, z be the Cartesian coordinates of a point in the image such that

x = distance to the right,

y = distance downward, and

z = distance forward of the midpoint between lenses.

---

1. This presentation is taken from NWL Report No. 1686, Subroutines for the NORC CRT Printer, U. S. Naval Weapons Laboratory, Dahlgren, Virginia, dated 9 March 1960.

2. Mirrors, Prisms, and Lenses, J. P. C. Southall (McMillan, N. Y. 1933), page 757.

Let x, y be the normal coordinates of the point in either view of the stereogram. Also, let

d = the interocular distance, and

u = the object distance of the lens.

And finally let

a = the displacement of the border from the optic axis, and

b = the breadth of the border.

For the left-hand view, the coordinate x is given by the equation,

$$x = \frac{1}{2} - \frac{a}{b} + \frac{u}{b} \frac{(x + \frac{1}{2} \; |d|)}{z} \qquad \text{LEFT}$$

For the right-hand view, the coordinate x is given by the equation,

$$x = \frac{1}{2} + \frac{a}{b} + \frac{u}{b} \frac{(x = \frac{1}{2} \; |d|)}{z} \qquad \text{RIGHT}$$

For both views, the coordinate y is given by the equation,

$$y = \frac{1}{2} + (1.2) \; \frac{u}{b} \; \frac{y}{z}$$

The parameters a, b, u may be present in the routines on such assumptions as

a. the field of view is 3 (ft.) wide at a distance away of 7 (ft.),

b. the breadth of border is 0.75 (in.) and the object distance is 1.75 (in.) and

c. the interocular distance is 2.5 (in.).

Whether the routine plots a left or right eye view is determined by the sign of d in the input. A left eye view is plotted for a negative value of d and a right eye view for a positive value of d. The point of projection is located midpoint between optic centers if the value of d is zero.

D. HISTOGRAMS

In constructing histograms, the programmer will find the subroutines LINRV, NONLNV and LINE2V of value.

The two routines LINRV and NONLNV - lower level modules employed by GRID1V are useful as building blocks for special grids such as histograms. LINRV may be used to generate only the vertical portion or the horizontal portion of a <u>linear</u> grid, while NONLNV generates only the vertical or horizontal portion of a <u>log</u> grid. These two routines are explained in Section II, page II-49.

LINE2V, which is used to draw a line from a fixed point in a specified direction, is explained in Section II, page II-78.

APPENDIX A: PACKAGE STRUCTURES

APPENDIX A

PACKAGE STRUCTURES

The following tables break down the various software packages to indicate the routines which are present in each package, as well as the number of locations, the entry points, and the external references for each routine. As modifications and changes are introduced into each system, the information contained within these tables may be altered slightly.

TABLE1: SCORS II, SCORS IV, LOCKHEED SYSTEMS

(Abbreviations - Loc. = number (octal) of locations, Ent. pts. = entry points, and Ext. ref. = external references)

Routine		SCORS II	SCORS IV	LOCKHEED
APLOTV (pg. II-71)	Loc.	221	272	204
	Ent. pts.	APLOTV	APLOTV	APLOTV
	Ext. ref.	SERREV SERSAV SCERRV NXV NYV (PLOT)	PLOTV SERSAV SERREV NXV NYV SCERRV SYSLOC	SCERRV NXV NYV PLOTV
APRNTV (pg. II-86)	Loc.	117	104	77
	Ent. pts.	APRNTV	APRNTV	APRNTV
	Ext. ref.	HOLLV PLOTV	HOLLV PLOTV SYSLOC	HOLLV PLOTV
BNBCDV[1] (pg. II-98)	Loc.	216	220	125
	Ent. pts.	BINBCD BNBCD	BINBCD BNBCD	BNBCD
	Ext. ref.	n/a	0.0 SYSLOC	n/a
BUFRV	Loc.	n/a	n/a	231
	Ent. pts.	n/a	n/a	BUFRV
	Ext. ref.	n/a	n/a	n/a

1. Routine called BNBCDV in Lockheed and SCORS II, called BNBCDQ in SCORS IV.

Routine		SCORS II	SCORS IV	LOCKHEED
CAMRAV[2] (pg. II-2)	Loc.	230	33	236
	Ent. pts.	CAMERA CAMRAV FRAMEV RESETV FORMV	CAMRAV SC1.. SC2.. SBC..	CAMRAV FRAMEV NOFRV FORMV RESETV BIGV SMALLV
	Ext. ref.	(PLOT) SCFL (CTJ..	CAMV.. PLOT.. Z..PTA	PLOT$
DSHLNV[3] (pg. II-80)	Loc.	546	535	257
	Ent. pts.	DOTLNV	DOTLNV	DSHLNV
	Ext. ref.	SQRT JDOT4V LINEV	.FXEM. LINEV JDOT4V SQRT SYSLOC	SQRT LINEV
DXDYV (pg. II-14)	Loc.	607	560	510
	Ent. pts.	DXDYV	DXDYV	DXDYV
	Ext. ref.	SETMOV	.FXEM. SETMOV SYSLOC	SETMOV ALOG10 NEXP5 NERR2
ERMRKV (pg. II-37)	Loc.	15	12	20
	Ent. pts.	ERMRKV	ERMRKV	ERMRKV
	Ext. ref.	(PLOT)	PLOT..	PLOT$

2. Routine called CAMRAV in Lockheed and SCORS IV, called CAMERA in SCORS II.
3. Routine called DSHLNV in Lockheed, DOTLNV in SCORS II, and DOTLNQ in SCORS IV.

Routine		SCORS II	SCORS IV	LOCKHEED
ERRLNV (pg. II-37)	Loc.	61	140	124
	Ent. pts.	ERRLNV	ERRLNV	ERRLNV
	Ext. ref.	ERMRKV	ERMRKV SYSLOC	ERMRKV
ERRNLV (pg. II-37)	Loc.	65	154	127
	Ent. pts.	ERRNLV	ERRNLV	ERRNLV
	Ext. ref.	ERMRKV	ERMRKV SYSLOC	ERMRKV
GRAC$	Loc.	(Various routines contained within GRAC$ Lockheed are separate routines of SCORS II and SCORS IV.)		105
	Ent. pts.			SETMIV SETMOV SETCIV SETCOV HOLDIV HOLDOV XMODV YMODV STOPTV BRITEV FAINTV PLOT$
	Ext. ref.			ALOG PLOT$
SETCIV	Loc.	20	10	
	Ent. pts.	SETCIV SETCOV	SETCIV SETCOV	These routines are contained in routine GRAC$; SETCIV through HOLDIV (inc.)
	Ext. ref.	n/a	WIDE..	

Routine		SCORS II	SCORS IV	LOCKHEED
SETMIV (pg. II-25)	Loc.	22	16	
	Ent. pts.	SETMIV SETMOV	SETMIV SETMOV	
	Ext. ref.	n/a	MB.. ML..	
STOPTV (pg. II-95)	Loc.	7	5	
	Ent. pts.	STOPTV	STOPTV	
	Ext. pts.	(PLOT)	PLOT.. STOP..	
BRITEV (pg. II-5) (pg. II-95)	Loc.	33	32	
	Ent. pts.	BRITEV FAINTV	BRITEV FAINTV	
	Ext. ref.	n/a	E.1 .XXSYY PLOT..	
XMODV[4] (pg. II-64)	Loc.	3	14	
	Ent. pts.	XMODV YMODV	XMODV YMODV	
	Ext. ref.	LOG	ALOG	
HOLDIV (pg. II-27)	Loc.	7	6	
	Ent. pts.	HOLDIV HOLDOV	HOLDIV HOLDOV	
	Ext. ref.	n/a	HOLD..	

4. Routine called XMODV in SCORS II, but called XMODQ in SCORS IV.

Routine		SCORS II	SCORS IV	LOCKHEED
GRID1V (pg. II-10)	Loc.	712	726	654
	Ent. pts.	GRID1V	GRID1V	GRID1V
	Ext. ref.	SERSAV SCERRV STOPTV BRITEV FRAMEV SETMOV SETCOV MSXYV HOLDOV ERRNLV ERRLNF XSCALV YSCALV NXV NYV NONLNV LINRV SERREV	SERCOV SETMOV BRITEV XSCALV SERSAV FRAMEV SERREV YSCALV ERRNLV ERRNLV LINRV NXV NYV MSXYV NONLNV HOLDOV SCERRV STOPTV SYSLOC	STOPTV BRITEV FRAMEV SETMOV SETCOV MSXYV HOLDOV ERRNLV ERRNLV XSCALV YSCALV NXY NYV NONLNV LINRV
HOLLV (pg. II-97)	Loc.	31	25	31
	Ent. pts.	HOLLV	HOLLV	HOLLV
	Ext. ref.	n/a	n/a	n/a
IDFRMV (pg. II-1)	Loc.	Contained in routine (PLOT)	Contained in routine (PLOT)	560
	Ent. pts.			IDFRM$ IDEND$ IDFRMV

Routine		SCORS II	SCORS IV	LOCKHEED
IDFRMV cont.	Ext. ref.			NOFRV FRAMEV FLAG$ RITE2V CHSIZV PLOT$
LABLV (pg. II-88)	Loc.	267	300	367
	Ent. pts.	LABLV	LABLV	LABLV
	Ext. ref.	BNBCDV PRINTV	BNBCDV PRINTV SYSLOC	PRINTV
LINEV (pg. II-78)	Loc.	133	241	130
	Ent. pts.	LINEV	LINEV	LINEV
	Ext. ref.	(PLOT)	.XXSYY SYSLOC PLOT..	PLOT$
LINRV (pg. II-49)	Loc.	632	657	555
	Ent. pts.	LINRV	LINRV	LINRV
	Ext. ref.	SERSAV SCERRV BNBCDV NXV NYV XAXISV YAXISV LABLV	.FXEM. SERSAV SERREV GRIDXY LINEV NXV NYV LABLV SCERRV BNBCDV SYSLOC	ALOG10 NXV NYV VLAGM LABLV NERR2$

Routine		SCORS II	SCORS IV	LOCKHEED
NONLNV (pg. II-49)	Loc.	434	437	365
	Ent. pts.	NONLNV	NONLNV	NONLNV
	Ext. ref.	SERSAV SCERRV BNBCDV LOG10 EXP(2 NXV NYV LABLV XAXISV YAXISV SERREV	.FXEM. .XP2. ALOG10 SERSAV SERREV GRIDXY LINEV NXV NYV LABLV SCERRV BNBCDV SYSLOC	ALOG10 NXV NYV LABLV VLAGM NEXP5$ NERR2$
POINTV (pg. II-75)	Loc.	144	142	176
	Ent. pts.	POINTV	POINTV	POINTV
	Ext. ref.	SERSAV NXV NYV PLOTV	NXV NYV PLOTV SERSAV SYSLOC	PLOTV NYV NXV
PLOTV (pg. II-81)	Loc.	26	17	20
	Ent. pts.	PLOTV	PLOTV	PLOTV
	Ext. ref.	(PLOT)	.XXSYY PLOT.. Z..PTA	PLOT$ SCCTAB
PLOT$[5] (pg. II-81)	Loc.	3077	775	260
	Ent. pts.	(PLOT) (CLEAN	PLOT.. PAGE40 NOFRV	PLOT$ CLEAN$ EOFTV

5. Routine is called PLOT$ in Lockheed package, PLOT.. in SCORS IV and (PLOT) in SCORS II.

Routine		SCORS II	SCORS IV	LOCKHEED
PLOT$ cont.		PLTND CRTAPE	4OUT CTPE.. PAGE 41	CAAINV RESIDV
		FRMNOV NOFRV	ID4G BIGV SMALLV	TPNUMV FLAG$
		(CTJ.. SCFL (JBN..	PLTND FRMNOV IDFRMV	
		TPNUMV IDFRAM		
		IDFRMV (P..4)		
	Ext. ref.	(IDS) (WRS) (WTC)	4020FL CAMV.. .OPEN	BUFRV$ NATB$
		(RCH) (TES) (WER)	.UN61. .WRITE SCFL	IDFRM$ IDEND$
		(RWT) RITE2V (TCO)	SYSDAT	TSWAP$
PRINTV (pg. II-85)	Loc.	134	110	122
	Ent. pts.	PRINTV	PRINTV	PRINTV
	Ext. ref.	(PLOT)	E.1 .XXSYY PLOT..	PLOT$ SCCTAB
			STOP.. SYSLOC	
RITE2V[6] (pg. II-99)	Loc.	242	256	260
	Ent. pts.	RITE2V RITSTV	RITE2V RITSTV RITXYV	RITE2V RITSTV
		RITXYV	HOLLV RTSIZV SYSLOC	RITXYV
	Ext. ref.	RTSIZV HOLLV	TABL1V VCHARV	TABL1V VCHARV
		VCHARV TAB1V		HOLLV RTSIZV

6. Routine called RITE2V in Lockheed and SCORS II, but RITE2Q in SCORS IV.

Routine		SCORS II	SCORS IV	LOCKHEED
SCCTAB	Loc.	n/a	n/a	100
	Ent. pts.	n/a	n/a	SCCTAB
	Ext. ref.	n/a	n/a	n/a
SCOUTV[7] (pg. II-119)	Loc.	156	177	145
	Ent. pts.	SCOUTV (BUMP (YTOP) (YREG) (SCRTV (CTAPE	SCOUTV BUMP..	SCOUTV
	Ext. ref.	(PLOT) (IOH)	CTPE.. FRAMEV .FFIL. .FWRD. .WRITE PLOT.. YREG..	FRAMEV RETSC$ PRINTV FRJ90 FORUT$ NTTAG$ FORSW$
SMXYV (pg. II-47)	Loc.	32	12	
	Ent. pts.	SMXYV MSXYV (SMXV) (SMYV)	SMXYV MSXYV	SMXYV MSXYV SMXV$ SMYV$
	Ext. ref.	(XXXX) (YYYY) XMODV YMODV	XXXX.. YYYY..	XMODV YYYY$ XXXX$

7. Routine called SCOUTV in Lockheed and SCORS II, but SCOUTQ in SCORS IV.

Routine		SCORS II	SCORS IV	LOCKHEED
VCHARV[8] (pg. II-105)	Loc.	247	245	260
	Ent. pts.	CHSIZV VCHARV RTSIZV	VCHX.. VCHY.. CHSIZV RTSIZV VCHARV	VCHARV CHSIZV RTSIZV
	Ext. ref.	(PLOT)	PLOT.. SYSLOC	PLOT$
VLAGM (pg. II-78)	Loc.	n/a	n/a	45
	Ent. pts.	n/a	n/a	VLAGM VLAGX VLAGY
	Ext. ref.	n/a	n/a	PLOT$
XAXISV (pg. II-54)	Loc.	52	51	22
	Ent. pts.	XAXISV YAXISV	XAXISV YAXISV	XAXISV YAXISV
	Ext. ref.	(PLOT)	GRIDXY LINEV	PLOT$
XSCALV[9] (pg. II-57)	Loc.	305	314	304
	Ent. pts.	XSCALV YSCALV NYV UXV UYV NXV SCERRV SERSAV SERREV (XXXX)	XSCALV YSCALV NXV NYV UXV UYV SCERRV SERRSAV SERREV SCLSAV RESCLV IXV IYV	XSCALV YSCALV SCERRV SCLSAV RESCLV NXV NYV IXV IYV UXV UYV

8. Routine called VCHARV in Lockheed and SCORS IV, but CHSIZV in SCORS II.
9. Routine called XSCALV in Lockheed and SCORS II, but XSCALQ in SCORS IV.

Routine		SCORS II	SCORS IV	LOCKHEED
XSCALV cont.	Ent. pts.	(YYYY) SCLSAV RESCLV IXV IYV XERRDD YERRDD		SERSAV SERREV
	Ext. ref.	(SMYV) (SMXV)	AX.. BX.. .HIGH. ....YY .LOW.. SYSLOC XERR.. XMODV XXXX.. YERR.. YMODV	SMYV$ SMXV$
ARGV	Loc.	64	n/a	n/a
	Ent. pts.	ARGV	n/a	n/a
	Ext. ref.	DUMP	n/a	n/a
BIGV (pg. II-5)	Loc.	12	Contained in PLOT..	Contained in CAMRAV
	Ent. pts.	BIGV SMALLV		
	Ext. ref.	(PLOT)		
JDOT4V[10]	Loc.	31	27	n/a
	Ent. pts.	JDOT4V INCRV	JDOT4V INCRV	n/a
	Ext. ref.	n/a	n/a	n/a

10. Routine called JDOT4V in SCORS II and INCRQ in SCORS IV.

Routine		SCORS II	SCORS IV	LOCKHEED
KWKPLT (pg. II-9)	Loc.	566	471	n/a
	Ent. pts.	KWKPLT	KWKPLT	n/a
	Ext. ref.	SERSAV SCERRV ARGV SMXYV DXDYV GRID1V APLOTV NXV NYV LINEV APPNTV PRINTV SERREV	APLOTV APRNTV DXDYV GRID1V LINEV NXV NYV PRINTV SCERRV SERREV SERSAV SMXYV SYSLOC	n/a
LINE2V[11] (pg. II-78)	Loc.	44	45	Contained in
	Ent. pts.	LINE2V	LINE2V	CAMRAV
	Ext. ref.	(PLOT)	PLOT..	
LOCSAV (pg. II-132)	Loc.	26	27	
	Ent. pts.	LOCSAV LOCSTV	LOCSAV LOCSTV	
	Ext. ref.	(YREG) (BUMP	BUMP.. REG..	
FORMV (pg. II-96)	Loc.	Contained in	6	Contained in
	Ent. pts.	CAMERA	FORMV	CAMRAV
	Ext. ref.		PLOT..	

11. Routine called LINE2V in SCORS II and LINE2Q in SCORS IV.

Routine		SCORS II	SCORS IV	LOCKHEED
FRAMEV (pg. II-3)	Loc.	Contained in CAMERA	43	Contained in CAMRAV
	Ent. pts.		FRAMEV RESETV STOP..	
	Ext. ref.		4020FL CAMV.. ID4G	
			PAGE40 PAGE41 PLOT..	
			SCFL	
SCPOOL	Loc.	n/a	50	n/a
	Ent. pts.		0.0 AX.. AY.. BX.. BY..	n/a
			NL.. MR.. MB.. MT..	
			XXXX.. YYYY.. .XX.YY	
			.XXSYY ....YY ...S..	
			Z..PTA WIDE.. HIGH..	
			CAMRV.. YTOP.. YREG..	
			HOLD.. .LOW.. .HIGH.	
			4020FL SCFL XERR.. YERR..	
	Ext. ref.	n/a	n/a	n/a
UN61	Loc.	n/a	1	n/a
	Ent. pts.	n/a	.UN61.	n/a
	Ext. ref.	n/a	n/a	n/a

Routine		SCORS II	SCORS IV	LOCKHEED
STOPTV (pg. II-95)	Loc.	n/a	5	n/a
	Ent. pts.	n/a	STOPTV	n/a
	Ext. ref.	n/a	PLOT.. STOP..	n/a
VCPS[12]	Loc.	12620	12440	n/a
(pg. II-135)	Ent. pts.	VCPS	VCPS	n/a
	Ext. ref.	TABL1V TABL2V	.FXEM. SCLSAV RITSTV	n/a
		TABL3V TABL5V	.FRDD. BRITEV PLOTV	
		TABL6V TABL7V	APRNTV RITE2V VCHARV	
		TABL8V TABL9V	XSCALV SERSAV FRAMEV	
		TAB11V TAB15V	SERREV YSCALV LINEV	
		TAB07V TAB05V	NXV NYV UXV UYV .FWRD.	
		SERSAV SCERRV	SCERRV HOLLV CASIZV	
		(RTN) (STH) (FIL)	.UNO2. .FCNV. .FRTN.	
		HOLLV NXV NYV	.UNO3. .FFIL. TABL3V	
		XSCALV YSCALV	TABL5V TAB15V TAB05V	
		SCLSAV UYV UXV	TABL6V TABL7V TAB07V	
		PLOTV SERREV	TABL8V TABL9V SYSLOC	
		FRAMEV LINEV	TABL1V	
		CHSIZV VCHARV		

12. Routine called VCPS in SCORS II and VCPQ in SCORS IV.

Routine		SCORS II		SCORS IV	LOCKHEED
VCPS cont.	Ext. ref.	APRNTV BRITEV			
		RITSTV RITE2V			
		(TSH)			
VINTV		LOC.	66	n/a	66
		Ent. Pts.	VINTV	n/a	VINTV
		Ext. Rep.	n/a	n/a	n/a

(This routine is a GSFC routine which exists with the SCORS II package for use on the IBM 7094. It is also included in the 1107/1108 system libraries).

Available tables and their amount of core

TABL1V	321	321	321
TABL2V	115	115	115
TABL3V	110	110	110
TABL5V	165	163	n/a
TABL6V	164	164	n/a
TAB L7V	107	107	n/a
TABL8V	134	134	n/a
TABL9V	127	127	n/a
TAB11V	327	327	n/a
TAB15V	330	330	n/a
TAB05V	165	165	n/a
TAB07V	207	207	n/a

TABLE 2: CSC PLOTTER PROGRAMMING SYSTEM

Routine	Size	Entry Points	External References
BOOK	766	BOOK JECT VANCE RESET CAMERA EXPAND REDUCE PLOTP PLOTH PLOTL PLOT I	SCCTAB COUNT$ RITE
BUFFER	167	BUFFER RITE	TTAB NTAB$
FIND	101	FIND	n/a
GRID	173	GRID	RITE
IDINFO	35	IDINFO	EDATE$ ETOD$
LABL2	172	LABL2	SET NNDU$ NI01$ NI02$
LABL3	27	LABL3	SET NWDU$ NI01$ NI02$
LABL4	115	LABL4	SET NWDU$ NI01$ NI02$
LETTER	666	LETTER	BOOK
LETTR2	666	LETTR2	BOOK
NBUFF$	445	NSET$ NLOC$ NEMP$ NSTB$	NFRZ$ NRGC$ NRGB$ NI01$
NDFILE	16	END FLE	TTAB NTAB$
NFINP	202	NRBU$ NRBT$	NABX NI02$ NLOC$ NFRZ$ NEMP$ NSET$ NERO$ NTAB$ NS11$ NSTB$

Routine	Size	Entry Points	External References
NFMT$	1046	NFMTS NT10$ NFRC$ NFGC$ FFGT$ NFBS$ NFB2$ NPCT$ NFCS$ NCS2$ NFRG$ TSTO TST1 NFAR$ NFRA$ NRTR$ NKLN$ NFDP$ NPW1$ NPW2$ NFC8$ NR91$ NDBLT$ FTGL NGC9$ NFB8$ NB90$ NCS9$ NP91$ NR92$ NRP3$ NFRH$	NI02$ NABZ$ NI02$ NI01$ NRGC$ NAB7$ NAB6$ NAB5$ NAB4$ NAB3$ NAB2$ NAB1$ NAB0$ NAVC$ NXVC$ NVEC$ NFRZ$
NFOUT	174	NWBT$ NWBU$	NABX$ NI02$ NLOC$ NFRZ$ NSET$ NERO$ NTAB$ NS11$ NSTB$
NFTV	23	NVEC$ NHVC$ NXVC$ NAVC$ NABO$ NAB1$ NAB3$ NAB4$ NAB5$ NAB6$ NAB7$	
NIER$	50	NERU$ NABX$ NABZ$ NIO1$ NI01$ NRGC$ N102$ N1O2$ NRGB$ NS11$ NFRZ$	
NINPT$	674	NRDC$ NRDU$ NRDT$	NDBLT$ NABZ$ FTGL NABX$ NERU$ NRGB$ TST1 NS11$

Routine	Size	Entry Points	External References
NIPT$ cont.			NTAB$ NVEC$ NI01$ NFC8$ NR91$ NKLN$ NFRA$ NRTR$ NFMT$ NFRH$ NFGT$ NFAR$ NR92$ NPW2$ NFRZ$ NFCS$ NCS9$ NCS2$ NP91$ NR93$ NFDP$ NPW1$ NB90$ NFBS$ NFRC$ NGC9$ NPCT$ NT10$ NFRG$ NFB8$
NOUT$	1510	NWDC$ NPRT$ NWDU$ NWDT$ PRCNG PLOTC NLETT SETBUF SETX SETMY SETSCL COUNT$ FLFPC SCCTAB	NFC8$ NABX$ NERU$ NRGB$ TSTO NS11$ NTAB$ NVEC$ NI01$ NFRH$ NR91$ NPCT$ NKLN$ NFRA$ NFMT$ NPW2$ NFAR$ NR92$ NFRZ$ NB90$ NFB2$ NR93$ NP91$ NPW1$ NFDP$ NFB8$ NFGT$ NFGC$ NT10$ NRTR$
NTAB$	101	NTAB$	
PLOT	2365	PLOT	PLOTID RITE VANCE LABL4 LABL3 LABL2 PLOTH PLOTVL

Routine	Size	Entry Points	External References
			PLOTVQ LETTER BUFFER
			SBUF FIND
PLOTID	65	PLOTID	IDINFO LETTER VANCE
PLOTVL	231	PLOTVL	BOOK
PLOTVQ	727	PLOTVQ	BOOK
PTYPE	157	PTYPE	COUNT$ RITE SCCTAB
REGNAM	12	REGNAM (Procedure)	
SBUF	63	SBUF	NEXP5
SET	35	SET	COUNT$ SETSCL SETMY
			SETMX SETY SETX SETBUF
			FLFPC NLETT$ LETTER

APPENDIX B: AVAILABLE ACCESSORY ROUTINES

APPENDIX B

AVAILABLE ACCESSORY ROUTINES

1. AVAILABLE UAIDE LIBRARY ROUTINES

At the GSFC, several UAIDE Library routines are available for use with the S-C 4020 Recorder. A list of these routines, with a brief description of each, follows. For more information concerning them, contact the Programming Methods Section, located in building no. 3, room 125.

Routine	Description
AICRT3 AICRT4	Subroutines which automatically plot data in graph format. AICRT4 is identical to AICRT3 but adds the capability for Hollerith array specification in title labeling. Each package is available in FORTRAN II and FORTRAN IV.
NAP	A set of Fortran and FAP coded routines consisting of three Fortran II chain links which process PERT output data to plot network charts on successive S-C 4020 frames.
APHLO	FAP program which produces a flow chart of a FORTRAN source program.
SCORS	Plotting subroutines which enable a wide variety of applications for the S-C 4020. This system is available for Fortran II and IV users and has been explained in Section II of this document.
MPLOT	Enables display of trajectories and orbits, as seen in perspective against earth, using XYZ coordinates on Shorline data tape. MPLOT requires SCORS package.
CSC Plotter Package for UNIVAC 1107/1108	The CSC Plotter Programming System has plot capabilities similar to the SCORS package but is not constructed parallel to it. The CSC Package, which provides flexibility in buffer areas and for writing to multiple plot tapes, is described in more detail in Section III of this document.

VCPS	Subroutine which plots matrix symbols, vector characters, vector lines, and justifies (but does not automatically hyphenate) lines. The SCORS package is required.
CESMAP	CESMAP draws a cylindrical equal-spaced map projection of the earth, centered about a specific longitude.
MERMEP	MERMAP draws a mercator map of the earth centered about a specific longitude.
YMERC	YMERC will deliver the Y-value to be plotted by MERMAP, if given the latitude of the point.
DRVEC (Fortran IV mapping package requires SCORS package and Latitude and Longitude Shoreline tape.)	DRVEC subroutine enables drawing of several vectors with one calling sequence.
LOCKHEED PACKAGE FOR UNIVAC 1107/1108	The Lockheed package closely resembles the SCORS package for the S-C 4020. This package is presented in Section II, along with the SCORS package.

2. PLOTTED CHARACTER INTENSITY: VINTV

This routine enables the programmer to control the intensity of plotted characters on the S-C 4020 Recorder according to a graded scale, from 1 to 16, designated in the calling sequence:

CALL VINTV (NC, IX, IY, INT)

where

NC	The character code (may be a decimal integer of position on grid or in form 1Hc.)
IX, IY	raster count coordinates of point to be plotted, and
INT	the intensity designator, coded in the range of 1 to 16 (light to dark).

This routine exists in FAP for use on the IBM 7094 with the SCORS II package. It is also available on the 1107/1108 system libraries.

It should be noted that VINTV is a one-shot routine. It does not reset the standard intensity (the S-C 4020 does not permit this), and all other S-C 4020 routines will be plotted at the existing intensity setting (high or low).

3. GENERALIZED PLOT ROUTINE: GFPLOT (including GCPLOT, GDPLOT, ENDPLT and IDPLOT)

GFPLOT is a series of routines which provide a rather basic but complete plotting system, designed to facilitate the programmer's problem of converting from computer to computer or plotter to plotter. In looking at GFPLOT, the programmer will see a close resemblance between it, APLOTV, and the Computer Science PLOT subroutine, explained in Section III. GFPLOT, however, provides the machine flexibility mentioned in the opening sentence.

In its design, GFPLOT is intended for use on the IBM 7094 or the UNIVAC 1107/1108 computers in order to produce output via the Stromberg-Carlson 4020 plotter, as well as the Calcomp 570 and the EAI 3440 plotters. Printer plotting is also available. Accompanying GFPLOT are the following routines: GCPLOT, GDPLOT, ENDPLT, and IDPLOT, all explained in this section. All routines are expected to be callable in FORTRAN IV, FORTRAN II, SLEUTH, FAP and MAP. However, as of this writing, all versions have not been implemented and their availability should be checked through the Programming Methods Section, building no. 3, room 125 at the GSFC.

GFPLOT: GFPLOT is the main entry and must be called prior to any call upon GCPLOT or GDPLOT. Its calling sequence is:

```
CALL GFPLOT (X, Y IDM, XMIN, XMAX, YMIN, YMAX, XLABLE,
YLABLE, ISYM, LINES , NAXES, NOFF)
```

where

X	Location of array containing floating point X coordinates.
Y	Location of array containing floating point Y coordinates.
IDM	Dimension of X and Y arrays.
XMIN	Minimum X value.
XMAX	Maximum X value.

YMIN	Minimum Y value.
YMAX	Maximum Y value.
XLABLE	Location of X-axis label of 12 dimension. (If word 1 of XLABLE is 0 the label will not be printed).
YLABLE	Location of Y-axis label of 12 dimension. (If word 1 of YLABLE is 0 the label will not be printed).
ISYM	Plotting symbol (integer); if less than zero, no symbol will be drawn.
LINES	Argument denoting the type of lines to be drawn and coded as

1 solid connecting lines between all on scale points,

2 solid connecting lines with no lines between points separated by off scale points,

3 solid connecting lines with dashed lines between points separated by off-scale points,

-2 dashed connecting lines with no lines between points separated by off-scale points,

-1 dashed connecting lines between all on-scale points, or

0 no connecting lines between points (overridden and all points connected by solid lines of a negative argument has been specified for ISYM)

NAXES Specifies the type of axis, coded as

-n a frame is drawn around the plot, where n is any non-zero integer

0 no frame or axes is drawn, or

2 a non-labeled axes is drawn.

NOFF Number of off-scale points encountered by GFPLOT (returned to calling program). This number will be printed on the righthand margin of the plot unless IDM is a negative quantity.

IDPLOT: The IDPLOT routine, which may be called at any time, prints an array of Hollerith information on the plot, center-adjusted with 72 characters to a line. The calling sequence is:

CALL IDPLOT (ID, IDDIM)

where

ID	An array containing Hollerith information
IDDIM	The dimension of the ID array, not to exceed 240.

GDPLOT and GCPLOT: These routines are used to plot additional information over a plot already drawn by GFPLOT. In general, GDPLOT will be used to change the plotting symbol. GCPLOT will vary according to the plotter being used - for the Calcomp and EAI plotters it will start a new plot over the old one, giving the operator a chance to change pens if a multicolored plot is desired; for the S-C 4020 plotter, it will change intensity from bright to dim; and for printer plots, GDPLOT and GCPLOT will operate identically. The calling sequence for each routine is identical.

CALL GxPLOT (X, Y, IDM, ISYM, LINES, NOFF)

where

All arguments in this call are identical to those of GFPLOT.

The maximum / minimum values of the previous GFPLOT will be used; the number of off-scale points will not be written on the plot, as in GFPLOT, but will be returned to the programmer.

ENDPLT: This is the final entry to the GFPLOT routine and must be called when all plotting is done to make sure that plot buffers are dumped, output tapes rewound and so forth. The calling sequence for ENDPLT is simply:

CALL ENDPLT

Any calls to the GFPLOT routine which follow a call to ENDPLT will be ineffective, i.e., an immediate return to the calling program will be initiated.

4. CURVE THROUGH SELECTED POINTS: QPLOTV

QPLOTV is an available subroutine which draws a curve through selected points via the S-C 4020 Plotter.

RESTRICTIONS This subroutine requires the use of the SCORS package on the IBM 7094 or the Lockheed package on the UNIVAC 1107/1108. It should be used only after scaling factors have been set up by GRID1V or by XSCALV and YSCALV.

Output is produced on the plot tape most recently assigned by a call to TPNUMV in a format readable by the S-C 4020, when TPNUMV is available.

The input parameters to QPLOTV should be in a monotonically increasing sequence. The curve will be drawn from the first point to the last. With respect to accuracy, the curve is made up of a sequence of straight lines which are between points 3 raster units apart in the X direction. The calling sequence:

CALL QPLOTV (NUM, XADD, YADD, X, Y)

where

NUM	Number of points in the curve (also the dimension of XADD, YADD, X and Y).
XADD, YADD	Specification of the coordinates of each point where the leftmost point is at XADD(1), YADD(1), and the rightmost at XADD(NUM), YADD(NUM).
X, Y	Are arrays of the dimension NUM which are used internally by QPLOTV for intermediate storage. They may be used elsewhere in the program but QPLOTV will destroy their contents.

5. ROUTINES TO STORE AND RETRIEVE PLOT COMMANDS: ONSAV, OFFSAV, GETSAV

At times it is necessary to generate a rather complex background which will be used repeatedly on many frames with different data points. It is desirable to save the plot commands which generate this background and retrieve them as needed rather than to go thru the plot routines which create these commands for each frame.

The series of routines needed to do this are:

CALL ONSAV (NUNIT)

NUNIT	FORTRAN logical unit on which to save plot commands; must be tape or drum, preferably drum. The plot commands generated by calls on the plot package following this call will be placed on NUNIT instead of the plot tape until a call is made of OFFSAV.

CALL OFFSAV	This call writes an end of file and rewinds NUNIT Calls on the plot package following this statement will again be written on the plot tape. NOTE: This may be followed by another CALL ONSAV using a different NUNIT to start saving plot commands for a different background.

CALL GETSAV (NUNIT)

NUNIT	As described for ONSAV. This call causes all the plot commands previously saved on NUNIT (an entire file) to be written on the plot tape. NUNIT is then rewound.

NOTE: THIS SERIES OF ROUTINES IS AVAILABLE ONLY FOR THE UNIVAC 1107/1108 AND IS CONTAINED ON THE AUXILIARY LIBRARY TAPE.

THE ELEMENT IS CALLED SAVERV.

APPENDIX C: ESTIMATES CONCERNING TAPE USAGE

APPENDIX C

ESTIMATES CONCERNING TAPE USAGE

The following tables and formulae are provided as aids to assist the programmer in computing tape consumption while plotting.

TIMING (IN MILLISECONDS):

Tape Drives	Low Density 200 cpi	High Density 556 cpi
IBM 729 IV	7.3 + .044N ms	7.3 + .017N ms
UNISERVO III C - VIIIC	6.0 + .044N ms	6.0 + .017N ms
IBM 729 IV	7.3 + .264K ms	7.3 + .102K ms
UNISERVO III C - VIIIC	6.0 + .264K ms	6.0 + .102K ms
N = no. of characters; K = no. of words. NOTE: There is a time reduction if there is no substantial wait between records.		

VOLUME:

Tape Drives	Low Density 200 cpi	High Density 556 cpi
IBM 729 IV	.75 + .005N in.	.75 + .0018N in.
UNISERVO III C - VIIIC	.75 + .005N in.	.75 + .0018N in.
IBM 729 IV	.75 + .030K in.	.75 + .0108K in.
UNISERVO III C - VIIIC	.75 + .030K in.	.75 + .0108K in.
N = no. of characters; K = no. of words.		

A. TOTAL WORDS IN IDENTIFICATION DATA FOR GRID (WID)

WID = VST + HST + TTL

where

VST = Total characters in vertical scale titles.

HST = Total characters in horizontal scale titles.

TTL = Total characters in top title lines.

B. TOTAL WORDS REQUIRED FOR ENTIRE GRID STRUCTURE (WIG)

WIG = NVG * WVG + NHG * WHG

where

NVG	=	No. of vertical grid lines.
WVG	=	Raster length of vertical grid line, divided by 64, plus 1.
NHG	=	No. of horizontal grid lines.
WHG	=	Raster length of horizontal grid lines, divided by 64, plus 1.

C. TOTAL WORDS REQUIRED FOR ALL GRID LABELS (WGL)

WGL = CVL * NVL + CHL * NHL

where:

CVL	=	No. of characters in vertical label (where constant).
NVL	=	No. of vertical labels.
CHL	=	No. of characters in horizontal label (where constant).
NHL	=	No. of horizontal labels.

D. TOTAL PLOT POINTS AND/OR LINE SEGMENTS TO PLOT ON THE GRID (WPL)

WPL = TPP + TLS

where:

TPP	=	Every plot point or character to place upon the grid as point plots.
TLS	=	Total line segments in any line graph.

E. TOTAL WORDS IN PICTURE (One or more graphs) (TWP)

TWP = WPL + WGL + WIG + WID

F. TOTAL WORDS PER JOB (TWJ)

TWJ = TWP * TNP

where:

TNP	=	Total no. of pictures (or frames).

G. TOTAL RECORDS IN JOB (TRJ)

$$TRJ = \frac{TWJ}{K}$$

where:

K = Words per record.

H. TOTAL TAPE TIME PER JOB (TTJ)

$$TTJ = TRJ\ (7.3 + .264K)$$

I. TOTAL LENGTH OF TAPE USED (Inches) (TLT)

$$TLT = TRJ\ (.75 + .030K)$$

J. TOTAL REELS OF TAPE (TRT)

$$TRT = TLT/\ 28440$$

APPENDIX D: TEST PROGRAMS AND SAMPLE PLOTS

APPENDIX D

TEST PROGRAMS AND SAMPLE PLOTS

TEST PROGRAM

The following is a sample test program which was written for compilation under FORTRAN IV IBSYS, using the SCORS IV package on the IBM 7094. The sample output, which immediately follows the program, is the resultant output on the S-C 4020 from the operation of the compiled sample program.

PROGRAM LISTING

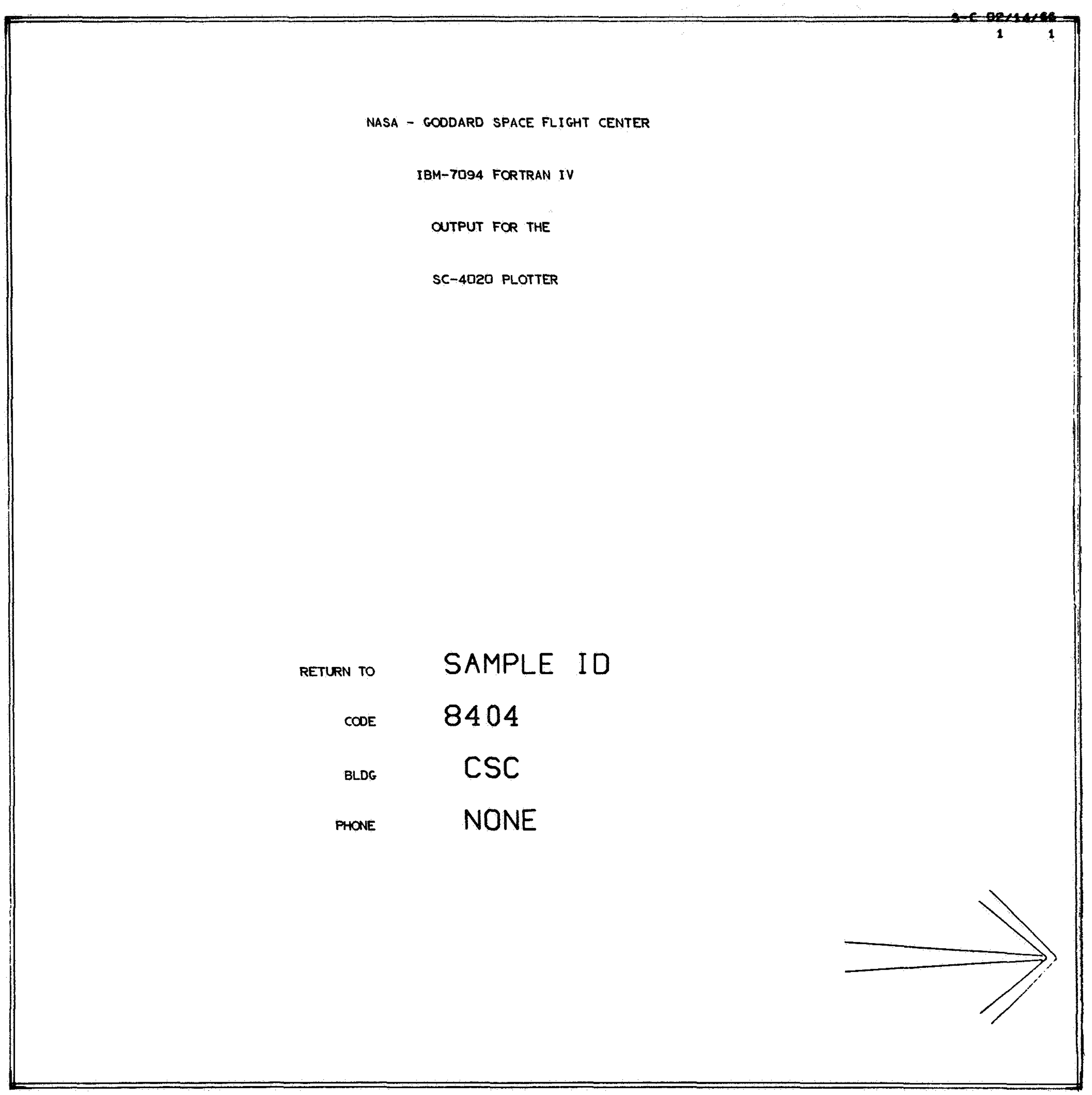
```
$IBSYS
$JOB        CO2T       ECS  5431  COMPUTER SCIENCES CORP.  588-8538
$PAUSE
$EXECUTE               IBJOB
$IBJOB                 MAP,GO
$IBFTC      ECS1       LIST,REF
      DIMENSION XX(10),YY(30),AMRK(3)
C     PLOTTING SYMBOLS --- A,B,C
      DATA (AMRK(I), I=1,3) /1HA,1HB,1HC/, THKNS/1.95/
C     X ARRAY IS STORED X1,X2, --- ,X10
      DATA  (XX(I),I=1,10)/2.5,4.5,6.5,8.5,10.5,12.5,14.5,16.5,18.5,20.5/
C     Y ARRAY IS STORED A1,A2, --- A10,B1,B2, ---, B10,C1,C2, ---, C10
      DATA  (YY(I),I=1,30) /10.5,12.4,14.4,16.5,19.0,21.3,24.2,27.6,31.0,
     X 37.0,18.0,21.2,24.4,27.6,31.0,34.7,38.9,43.9,49.9,57.7,26.1,
     X 30.8,35.8,40.9,46.2,51.8,58.2,65.8,75.2,87.1/
      CALL IDFRMV (9HSAMPLE ID, 4H8404,6H  CSC   ,6H  NONE  )
C     THIS SELECTS BOTH CAMERAS
      CALL CAMRAV (916)
C     THESE LOOPS DETERMINE MINIMUM AND MAXIMUM X OR Y FOR USE IN
C     SCALING
      XL=XX(1)
      XR=XX(1)
      DO 5 I=2,10
      XL=AMIN 1(XL,XX(I))
      XR=AMAX1(XR,XX(I))
    5 CONTINUE
      YB=YY(1)
      YT=YY(1)
```

```
        DO  10  I=2,30
        YB=AMIN1 (YB,YY(I))
        YT=AMAX1(YT,YY(I))
   10   CONTINUE
C    CALCULATE X AND Y SCALES
        CALL DXDYV (1,XL,XR,DX,N,I,NX,10.0,IERR)
        CALL DXDYV (2,YB,YT,DY,M,J,NY,10.0,IERR)
C    ADVANCE FRAME AND OUTPUT SCALED AND LABELED GRID
        CALL GRID1V  (1,XL,XR,YB,YT,DX,DY,N,M,I,J,NX,NY)
        CALL APRNTV(0,-12,-14,14HTEMPERATURE  F ,4,570)
        CALL PRINTV  (-10,10HTIME  (SEC) ,472,4)
        CALL PRINTV  (-10,10HTHICKNESS=  ,200,1015)
C    CONVERT A FLOATING POINT NUMBER TO BCD, AND OUTPUT
        CALL LABLV (THKNS,296,1015,5,1,2)
C    OUTPUT THE DATA POINTS
C       X1  A1,B1,C1
C       X2  A2,B2,C2
C       ETC
        DO  15 I=1,10
        CALL APLOTV  (30,XX(I),YY(I),0,10,3,AMRK,IERR)
   15   CONTINUE

C    NOW CONNECT DATA POINTS
     DO 60 I=1,3
     J=(I=1)*10
     DO 50 K=1,10
     IY=J+K
     IX2=NXV(XX(K))
     IY2=NYV(YY(IY))
     IF (K-1) 45,45,20
C    NOTE THAT ZERO USUALLY INDICATES AN OFF-SCALE DATA POINT
C    IN PRACTICAL PROBLEMS, A MORE COMPLETE ERROR PROCEDURE
     IS SUGGESTED
     20  IF (IX2*IY2) 21,21,25
     21  WRITE (6,200) I,K
    200  FORMAT  (31 HOFF SCALE DATA POINT X(I,J), I=12,3H J=,12)
         GO TO 45
C    ALL A'S ARE CONNECTED BY USE OF LINEV
C    B'S AND C'S USE DSHLNV
     25 GO TO (30,35,40),I
     30 CALL LINEV (IX1,IY1,IX2,IY2)
         GO TO 45
     35 CALL DOTLNV (IX1,IY1,IX2,IY2,8,2)
         GO TO 45
     40 CALL DOTLNV (IX1,IY1,IX2,IY2,5,5)
```

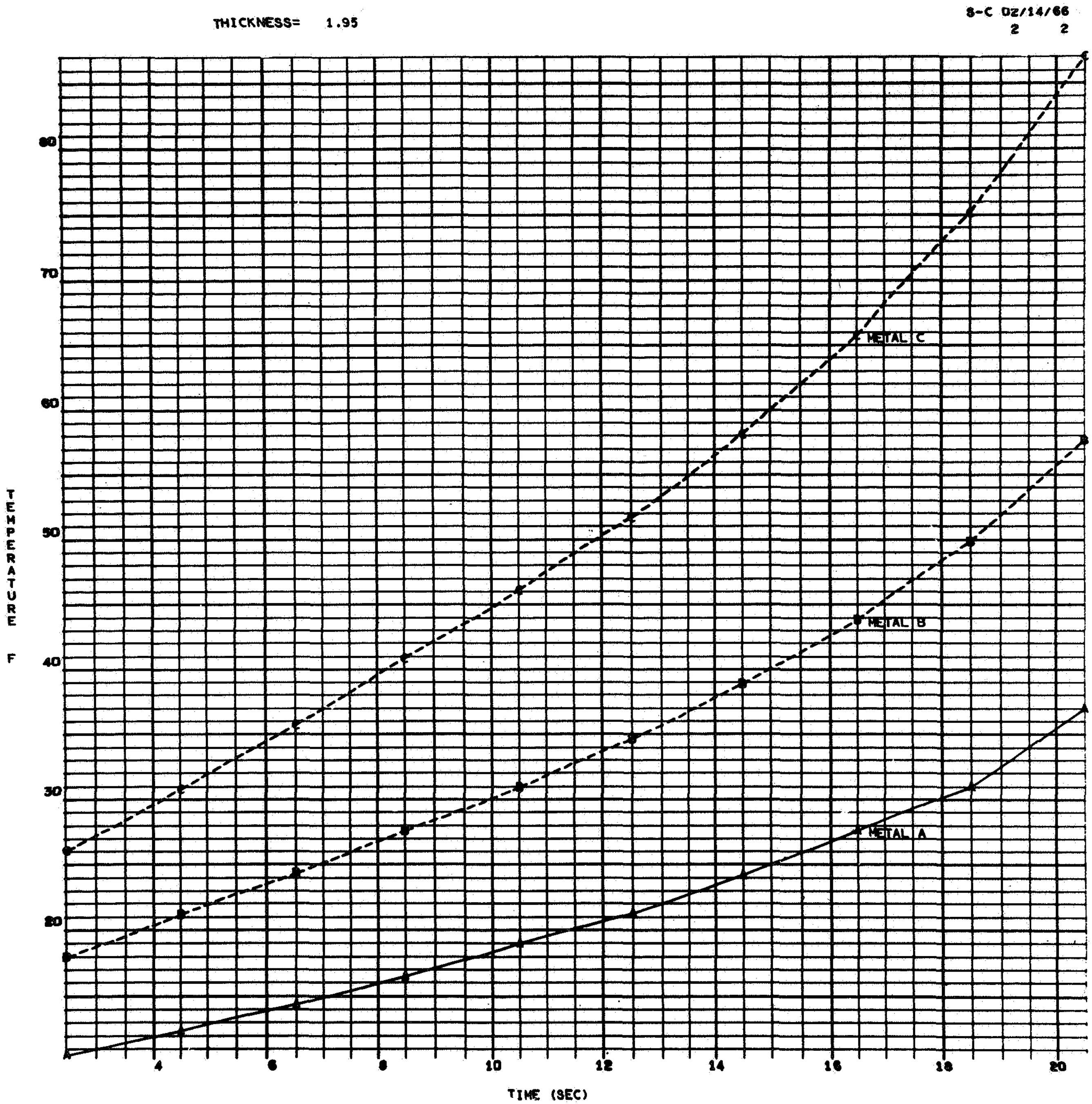
```
45    IX1=IX2
      IY1=IY2
50    CONTINUE
60    CONTINUE
      NX1=NXV(XX(8))+14
      NY1=NYV(YY(8))-3
      NY2=NYV(YY(18))-3
      NY3=NYV(YY(28))-3
C   ANNOTATE EACH TRACE
      CALL PRINTV (-7,7HMETAL A,NX1,NY1)
      CALL PRINTV (-7,7HMETAL B,NX1,NY2)
      CALL PRINTV (-7,7HMETAL C,NX1,NY3)
C   LEAVE A CLEAR FRAME (NO CORNERS, NO ID)
      CALL FRAMEV (3)
      CALL PLTND
      END FILE 7
      STOP
      END
```

SAMPLE OUTPUT

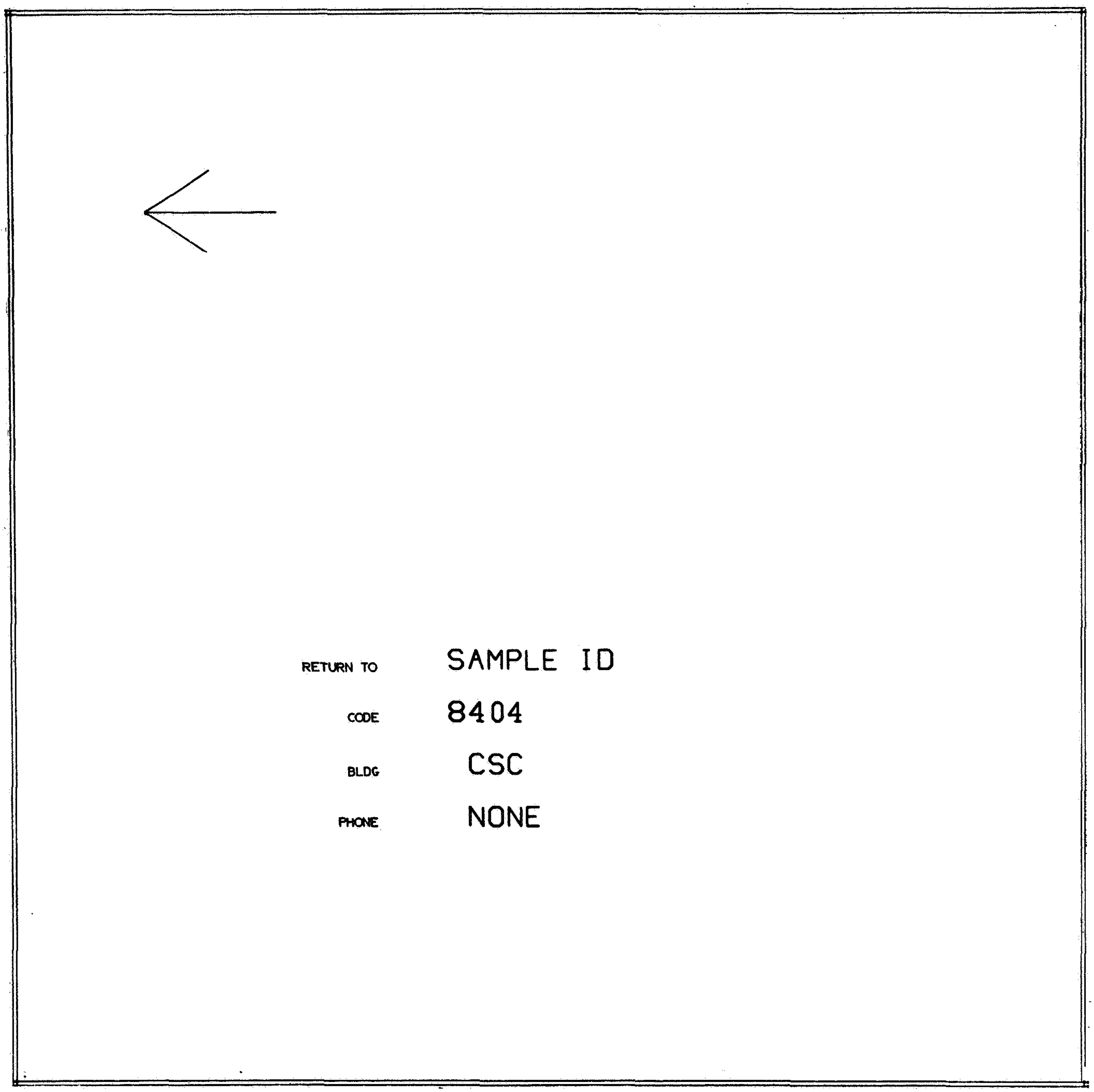
The following three pages show the output produced by running the previous program. This output consists of the initial ID frame, the plotted graph, and the trailing ID frame.



Identification Frame (initial)



Sample Output Plot



Identification Frame (trailing)

OTHER PACKAGES

As stated, the previous program was FORTRAN IV oriented. In order to compile this program under SCORS II, the data must be defined according to FORTRAN II rules of definition(eg. DATA statement is not applicable under FORTRAN II). Otherwise, the calling sequences themselves are correct.

As for an 1107/1108 compilation under Lockheed package, it must be noted that to call routine DSHLNV, the call must be for DSHLNV and not DOTLNV as with the SCORS packages. The routine EOFTV is called to terminate the job.

1107 Identification Frame (initial)

NASA - GODDARD SPACE FLIGHT CENTER

UNIVAC 1107 OUTPUT

FOR THE

SC4020 MICROFILM RECORDER

****ATTENTION, PROGRAMMER****

THE 1107 DOES NOT HAVE INTERNAL INFORMATION FOR COMPLETING THIS ID FRAME. PLEASE INSERT THE FOLLOWING STATEMENT INTO YOUR PROGRAM PRIOR TO ANY OTHER SC4020 SUBROUTINE CALL (INCLUDING CAMRAV).

```
CALL IDFRMV (NAME,CODE,BLDG,PHONE)
```

ALL ARGUMENTS MUST BE HOLLERITH DATA, I.E., DEFINED WITH H FORMAT OR READ IN WITH A FORMAT. THE PROGRAM REQUIRES TWO WORDS (6 CHARACTERS EACH) FOR NAME PLUS ONE WORD FOR EACH OTHER ARGUMENT. A SAMPLE FOLLOWS.

```
CALL IDFRMV (11HJ J JOHNSON,4H543M,6HBLDG 3,4H5809)
```

OUTPUT OF THIS EXPLANATION WILL BE SUPPRESSED IF IDFRMV IS CALLED.

RETURN TO NAME

CODE CØDE

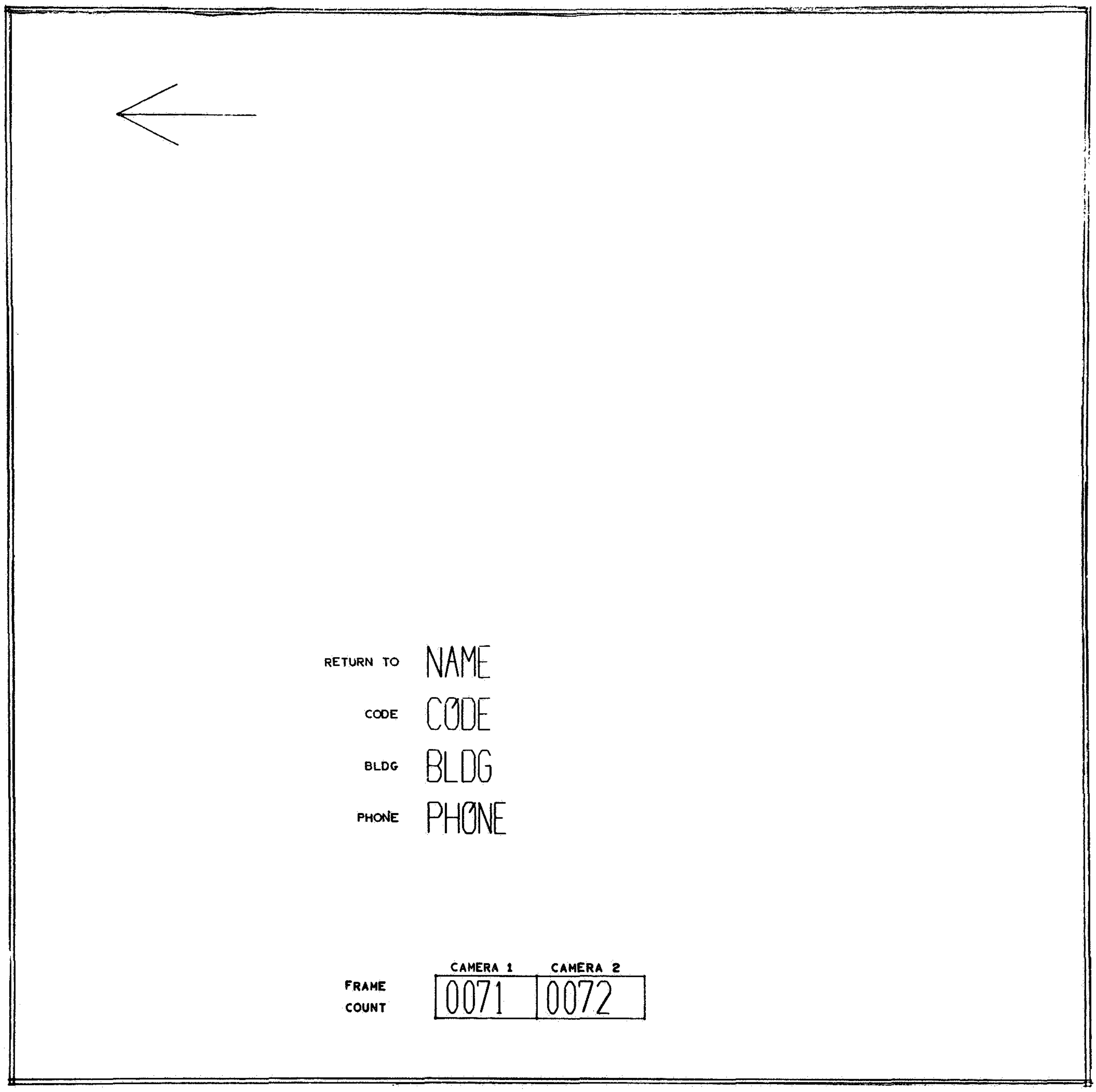
BLDG BLDG

PHONE PHØNE

Note the request to the programmer to use IDFRMV to insert identification data.

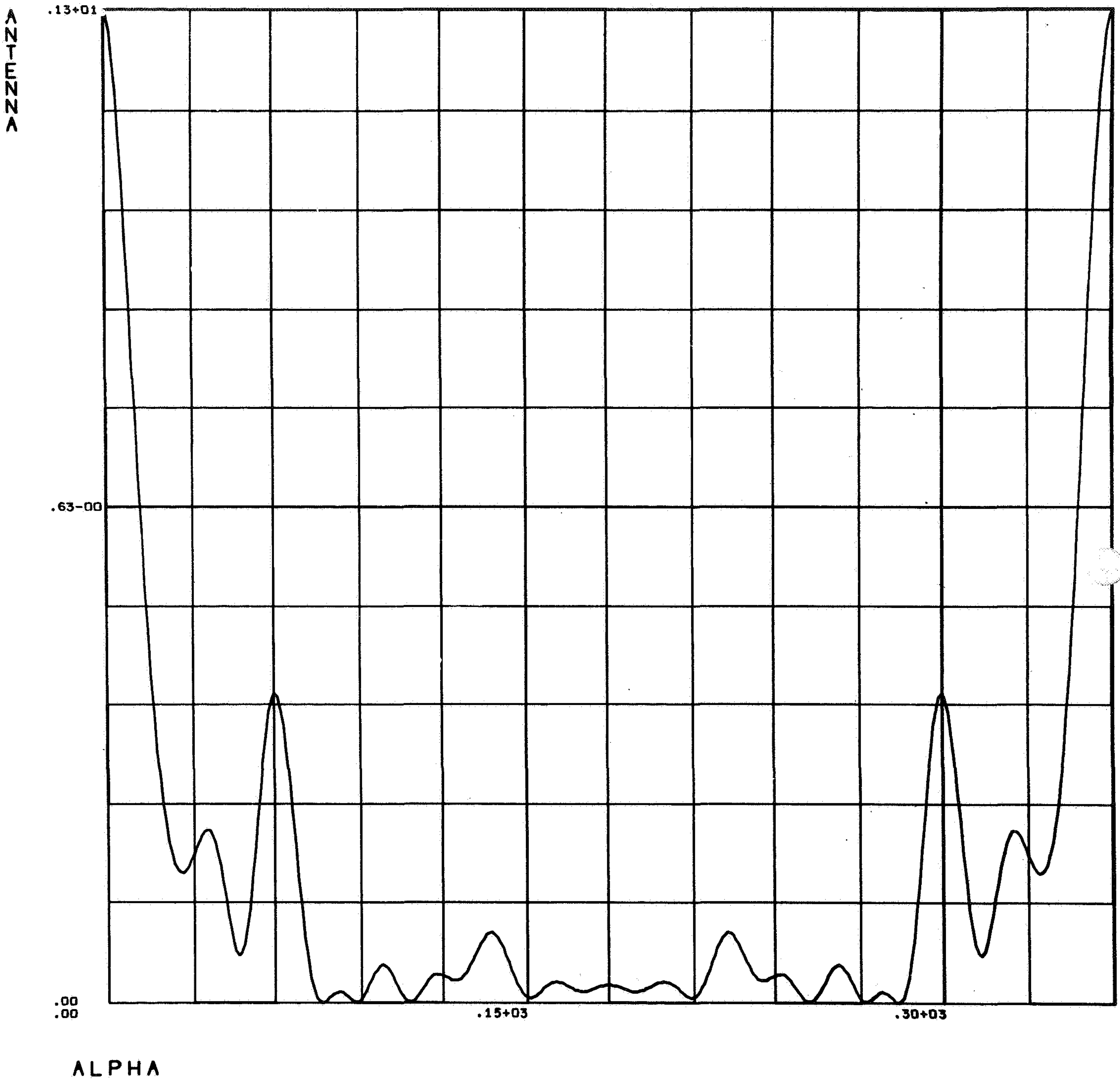
Trailing identification frame on next page.

1107 Identification Frame (trailing)



Again, there is no identification data (failure to use IDFRMV), but frame count for each camera appears.

Sample CSC Plot Output



CALL PLOT (J, ALFD, A, 2HLH, 0., 310., 0., AAMMX, 30., AINC, 5HALPHA, 7HANTENNA).

APPENDIX E: GSFC FORM SLIDES

APPENDIX E

This Appendix illustrates and describes the various Form Slides which are available for use with the S-C 4020. Refer to the text for programming information regarding the use of the form flash feature. These slides are kept by the S-C 4020 operator, and may be requested for use on a given S-C 4020 run by means of the identification given with each of the following descriptions (for example, PMS-1, PMS-2, etc.).

It bears repeating that registration and alignment of the slide with respect to the S-C 4020 plotting area cannot be fully guaranteed. Repeatability after a slide has been removed and reinserted (as will occur from run to run) may not produce precisely the same orientation. Shifts of as much as 10 raster counts, and skewing of up to 5 raster counts from one side of the frame to the other have been observed. Distortion of the slide image to a barely discernable extent may occur. This is evidenced by a slight curvature of straight lines and slight magnification of scale from one edge of the slide image to the opposite edge. For these reasons, the scaling information and formulae given in the following descriptions must be treated as approximations.

The S-C 4020 operator may be instructed to rotate the Charactron tube prior to doing a run on the plotter. When also using Form Slides, the slide image does not rotate with respect to the film (or paper), but plotting which is produced on the tube does.

The illustrations which follow were produced with the "normal" setting of Charactron tube rotation. The legend in the upper right corner and the four corner marks in each figure were produced by software on the Charactron tube. The remainder of each figure is the slide image.

PMS-1

This slide, illustrated in Figure E-1, is a Mercator projection world map. In addition to the borderlines, grid lines every 30° in both latitude and longitude appear. Note that the identification, PMS-1, appears in the lower left corner of the map. Horizontally, the map image extends from 65°E through Greenwich (0°) to 105°E longitude. This is more than once around the globe so that a portion of it ($65^{\circ}E \leq \lambda \leq 105^{\circ}E$) is repeated on both left and right sides. The vertical scale extends from 70°S to 84°N latitude. It does <u>not</u> include the poles, and the point $\varphi = 0^{\circ}$ is not centered in either direction.

In order to overplot data which are available in terms of latitude and longitude, they must first be rescaled. The horizontal scale is linear; the vertical scale is not. Given λ and φ in degrees ($-70 \leq \varphi \leq +84$ and $-180 \leq \lambda \leq +180$), approximate conversion formulae to obtain X and Y in raster counts are:

$$X = 2.45\lambda + 742 \qquad \text{for } -180 \leq \lambda \leq +105$$

$$X = 2.45\lambda - 140.25 \qquad \text{for } +65 \leq \lambda \leq +180$$

(Note: Both formulae for X may be used in the range $+65 \leq \lambda \leq +105$ to plot the same point twice in the duplicated area of the map.)

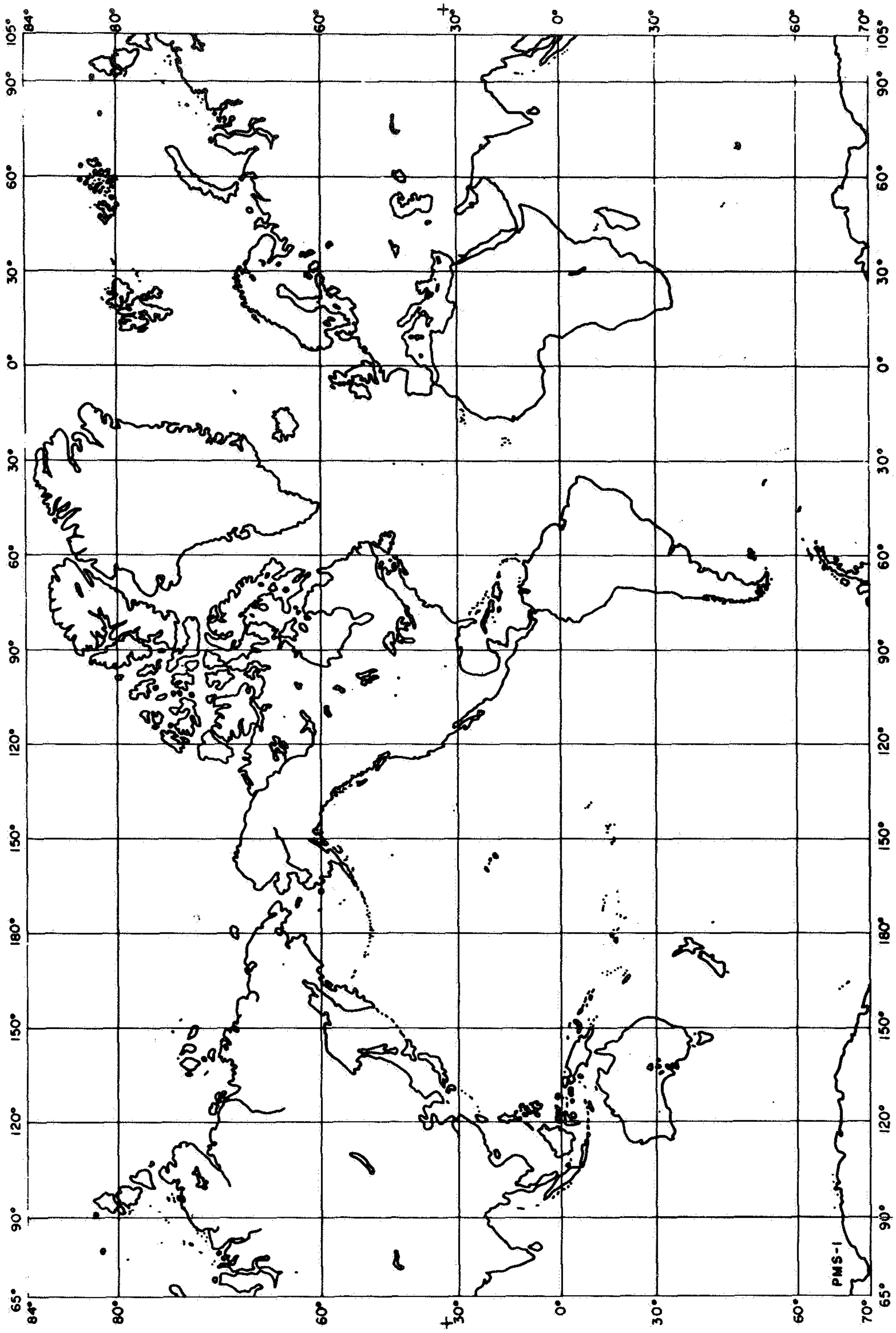
$$Y = 427 - 322.71 \log_{10}\left[\operatorname{Tan}\left(\frac{90^{\circ} - \varphi}{2}\right)\right]$$

$$= 427 - 140.15 \ln\left[\operatorname{Tan}\left(\frac{90^{\circ} - \varphi}{2}\right)\right]$$

The following information is given in order to assist in positioning title lines or other data which the user may wish to display:

$\lambda = 65^{\circ}$E occurs at approx. 19 raster counts; 0° at 742; 105° at 999.

$\varphi = 70^{\circ}$S occurs at approx. 186 raster counts; 60°S at 243; 30°S at 351; 0° at 427; 30°N at 504; 60°N at 613; 80°N at 722 and 84°N at 843.



(This Figure has been Rotated 90° to the Left for Printing Purposes Only)

Figure E-1

PMS-2

This slide, illustrated in Figure E-2, is a Miller Cylindrical projection world map. Grid lines appear every 10° in both latitude and longitude. Note that the identification, PMS-2, appears in the lower left corner of the map. The horizontal scale extends from 180°W to 180°E longitude and is linear. The vertical scale extends from 90°S to 90°N latitude (pole to pole) and is not linear. The point $\varphi = 0^{\circ}$, $\lambda = 0^{\circ}$ is centered on the map.

In order to overplot data which are available in terms of latitude and longitude, they must first be rescaled. Given φ and λ in degrees ($-90 \leq \varphi \leq +90$ and $-180 \leq \lambda \leq +180$), approximate conversion formulae to obtain X and Y in raster count are:

$$X = 2.70\ \lambda + 508$$

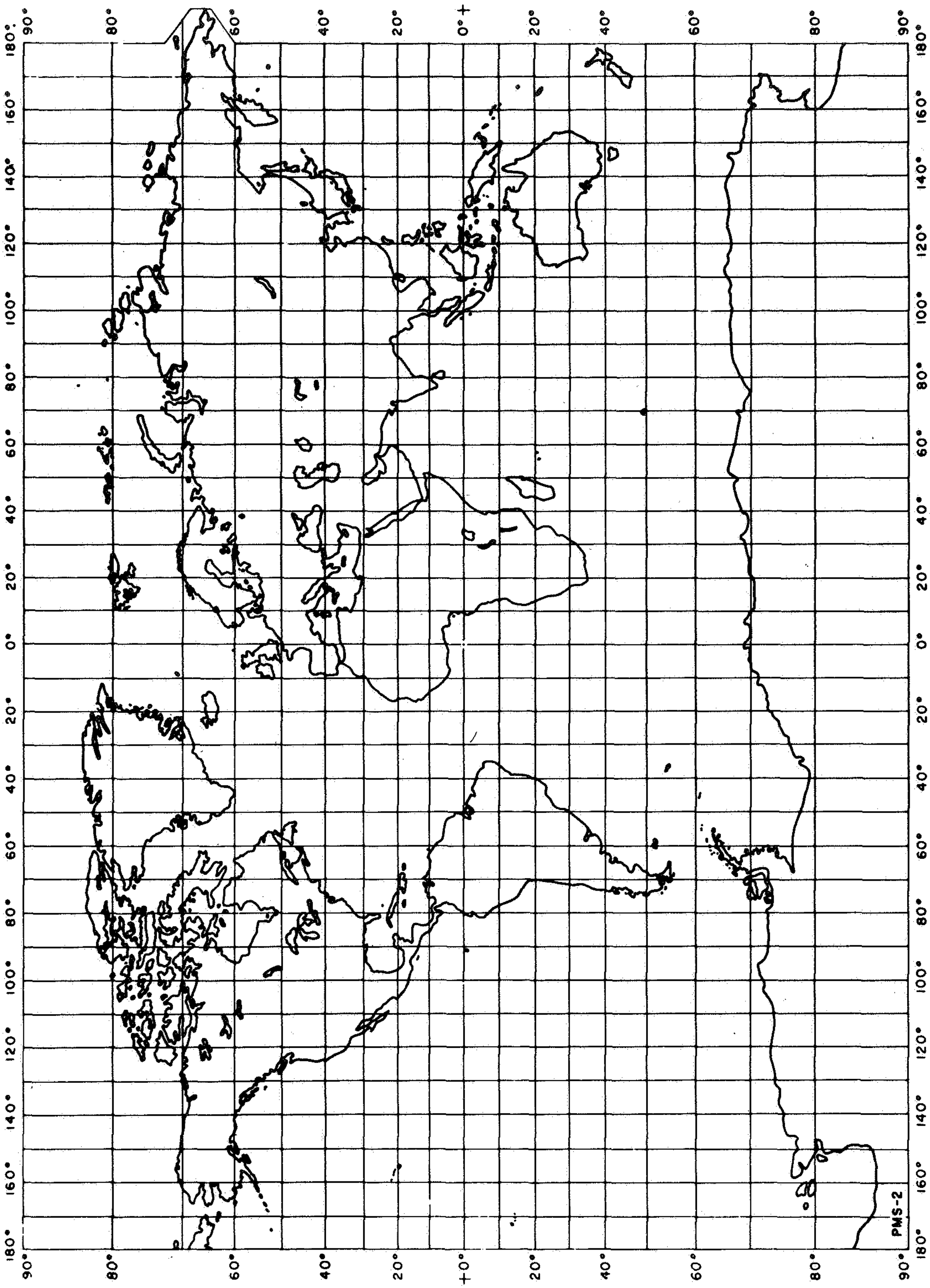
$$Y = 444.67 \log_{10} \text{Tan}\ (45^{\circ} + 0.4\varphi) + 514$$

$$= 193.12 \ln \text{Tan}\ (45^{\circ} + 0.4\varphi) + 514$$

The following information is given in order to assist in positioning title lines or other data which the user may wish to display:

λ $= 180^{\circ}$W occurs at approximately 22 raster counts; 0° at 508; 180°E at 994.

φ $= 90^{\circ}$S occurs at approximately 160 raster counts; 80°S at 232; 40°S at 400; 0° at 514; 40°N at 628; 80°N at 798; 90°N at 870.



(This Figure has been Rotated 90° to the Left for Printing Purposes Only)

Figure E-2

PMS-3

This slide, illustrated in Figure E-3, is a Miller Cylindrical projection world map. It differs from PMS-2 in two respects: Rather than a complete grid, there are tick marks every 10° along each border, and the map is rotated 90° counter-clockwise with respect to PMS-2. With normal positioning of the Charactron tube, positive Y is the positive (East) longitude direction and positive X is the negative (South) latitude direction. The longitude scale extends from 180°W to 180°E and is linear; The latitude scale extends from 90°S to 90°N (pole to pole) and is not linear. The point $\varphi = 0^{\circ}$, $\lambda = 0^{\circ}$ is centered on the map.

In order to overplot data which are available in terms of latitude and longitude, they must first be rescaled. Given φ and λ in degrees ($-90 \leq \varphi \leq +90$ and $-180 \leq \lambda \leq +180$), approximate conversion formulae to obtain X and Y in raster counts are:

$$X = 510 - 446.92 \log_{10} \mathrm{Tan}(45^{\circ} + 0.4\varphi)$$

$$= 510 - 194.09 \ln \mathrm{Tan}(45^{\circ} + 0.4\varphi)$$

$$Y = 519 - 2.68 \lambda$$

The following information is given in order to assist in positioning title lines or other data which the user may wish to display.

λ $= 180^{\circ}$W occurs at approximately 37 rasters counts; 0° at 519; 180°E at 1002.

φ $= 90^{\circ}$S occurs at approximately 866 raster counts; 80°S at 793; 40°S at 625; 0° at 510; 40°N at 395; 80°N at 225; 90°N at 153.

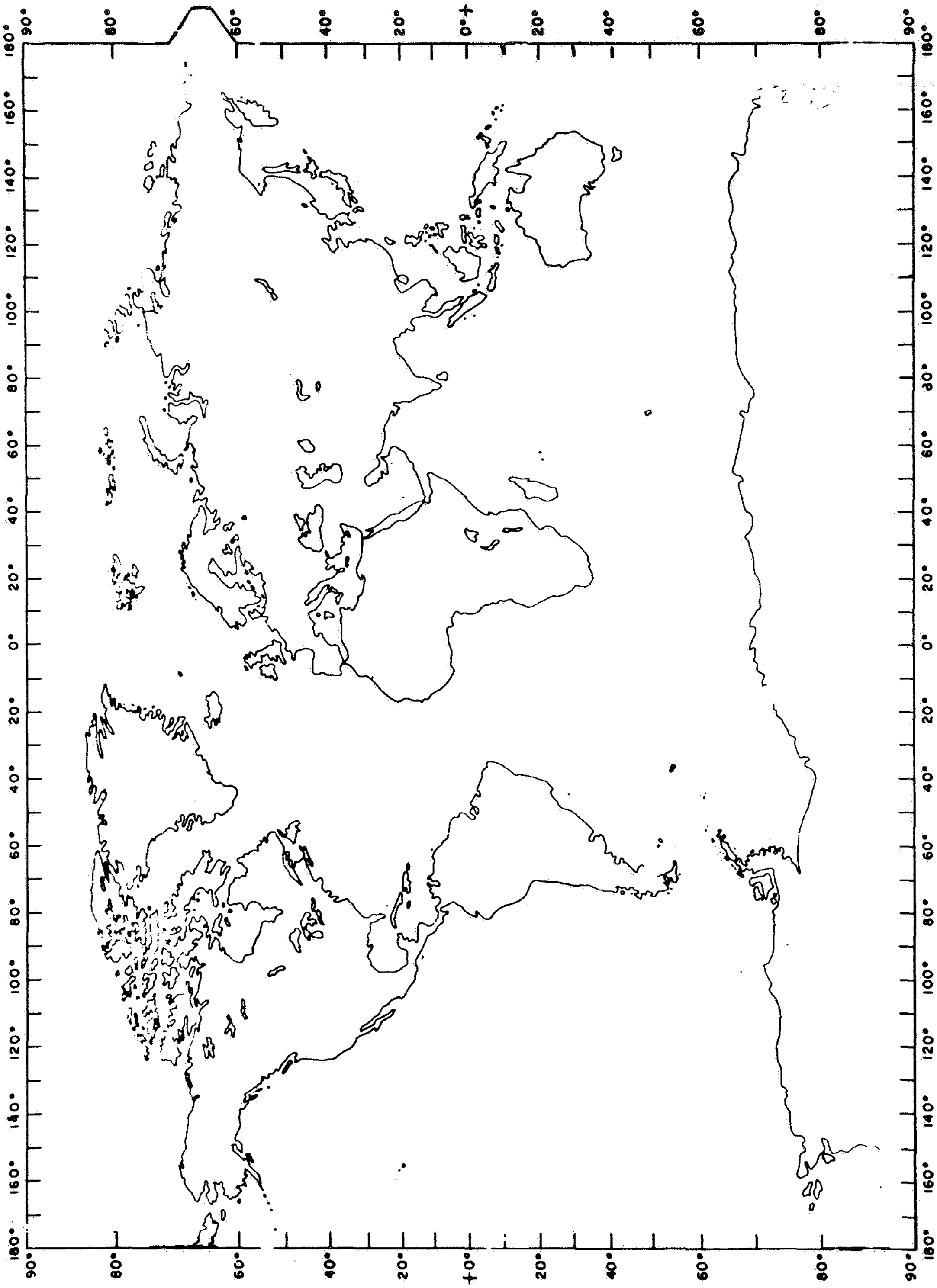

90°
80°
60°
40°
20°
0°
20°
40°
60°
80°
90°
180°
160°
140°
120°
100°
80°
60°
40°
20°
0°
20°
40°
60°
80°
100°
120°
140°
160°
180°

Figure E-3

PMS-4

This slide, illustrated in Figure E-4, is a world map characterized by a linear latitude scale. Land masses are shaded and somewhat stylized, and grid lines appear every 10° in both latitude and longitude. The map is oriented such that, with normal positioning of the Charactron tube, the programmer's positive Y is the positive (East) longitude direction and positive X is the negative (South) latitude direction. The longitude scale extends from 300°W (=60°E) through Greenwich (0°) to 60°E. The latitude scale extends from 90°S to 90°N (pole to pole). Both scales are linear. The point $\varphi = 0^{\circ}$, $\lambda = 0^{\circ}$ is not centered on the map.

In order to overplot data which are available in terms of latitude and longitude, they must first be rescaled. Given φ and λ in degrees ($-90 \leq \varphi \leq +90$ and $-180 \leq \lambda \leq +180$), approximate conversion formulae to obtain X and Y in raster counts are:

$$X = 508 - 4.16\,\varphi$$

$$Y = 2.80\,\lambda - 146 \qquad \text{for } +60 \leq \lambda \leq +180$$

$$= 2.80\,\lambda - 863 \qquad \text{for } -180 \leq \lambda \leq +60$$

The following information is given in order to assist in positioning title lines or other data which the user may wish to display:

λ = 300°W (left margin) occurs at approximately 22 raster counts; 200°W at 301; 120°W at 525; 40°W at 751; 0° at 863; 60°E (right margin) at 1030 (which is off scale).

φ = 90°S occurs at approximately 882 raster counts; 80°S at 840; 60°S at 757; 30°S at 632; 0° at 508; 30°N at 382; 60°N at 258; 90°N at 134.

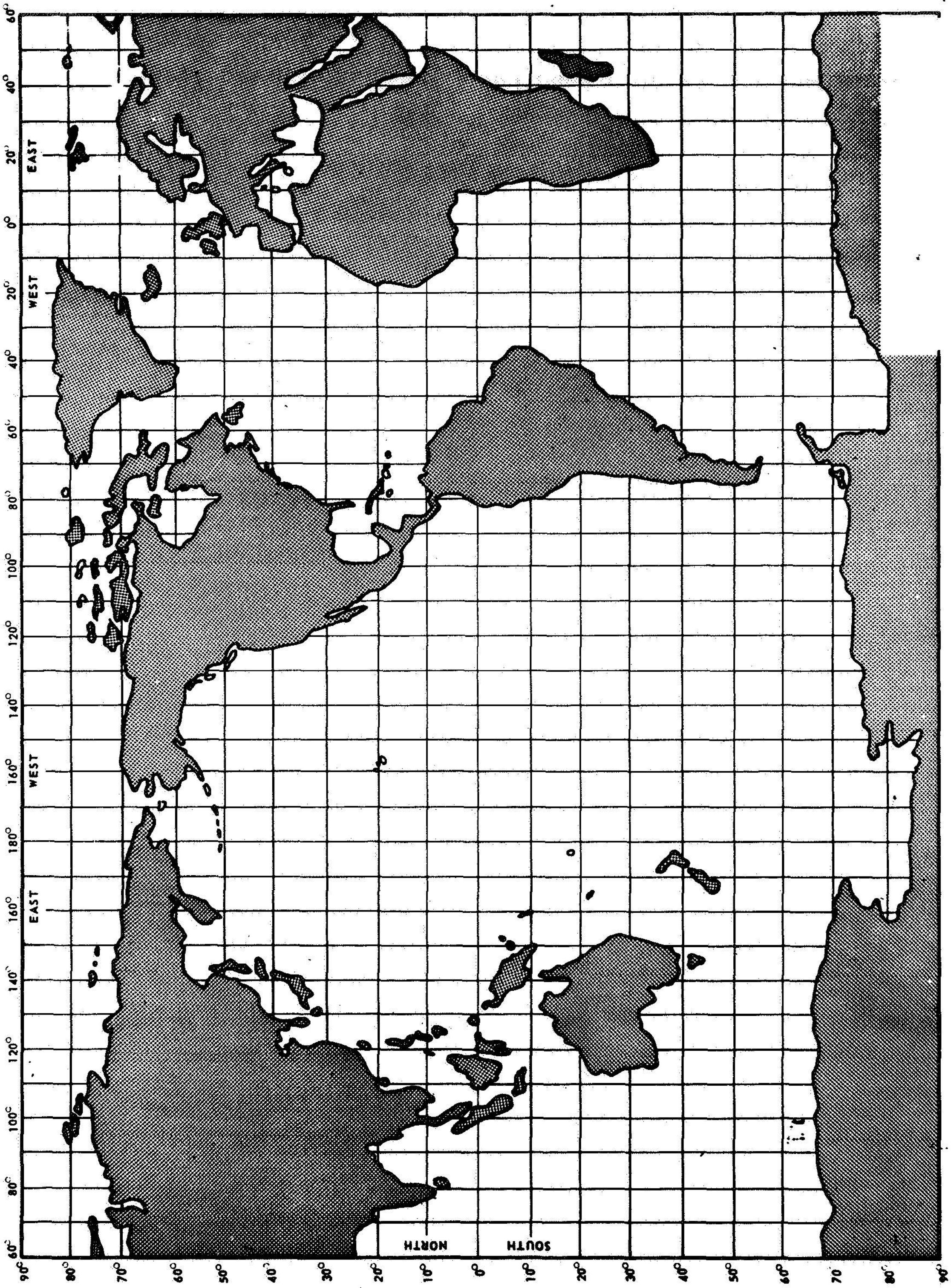


Figure E-4

PMS-5

This slide, illustrated in Figure E-5, is a Miller Cylindrical projection strip map. Two strips are displayed and on each, the Western hemisphere is shown three times and the Eastern hemisphere twice. Grid lines appear every 10^{o} in both latitude and longitude. The latitude scale ranges from 85^{o}S to 85^{o}N (does not include the poles) and is not linear.

In order to overplot data which are available in terms of latitude and longitude, they must first be rescaled. Due to the recycling of the longitude scale, an equation is not given here for determining X in raster counts as a function of λ. Instead, the programmer may position himself as desired with the following information. There are 1.09 raster counts per degree of longitude. The first (leftmost) occurrence of $\lambda = 180^{o}$ is at approximately 18 raster counts, the second at 409, the third at 804. The first occurrence of $\lambda = 0^{o}$ is at approximately 212 raster counts, the second at 607, the third at 999. Given φ in degrees ($-85 \leq \varphi \leq +85$), the approximate value of Y in raster counts may be determined from the following formulae:

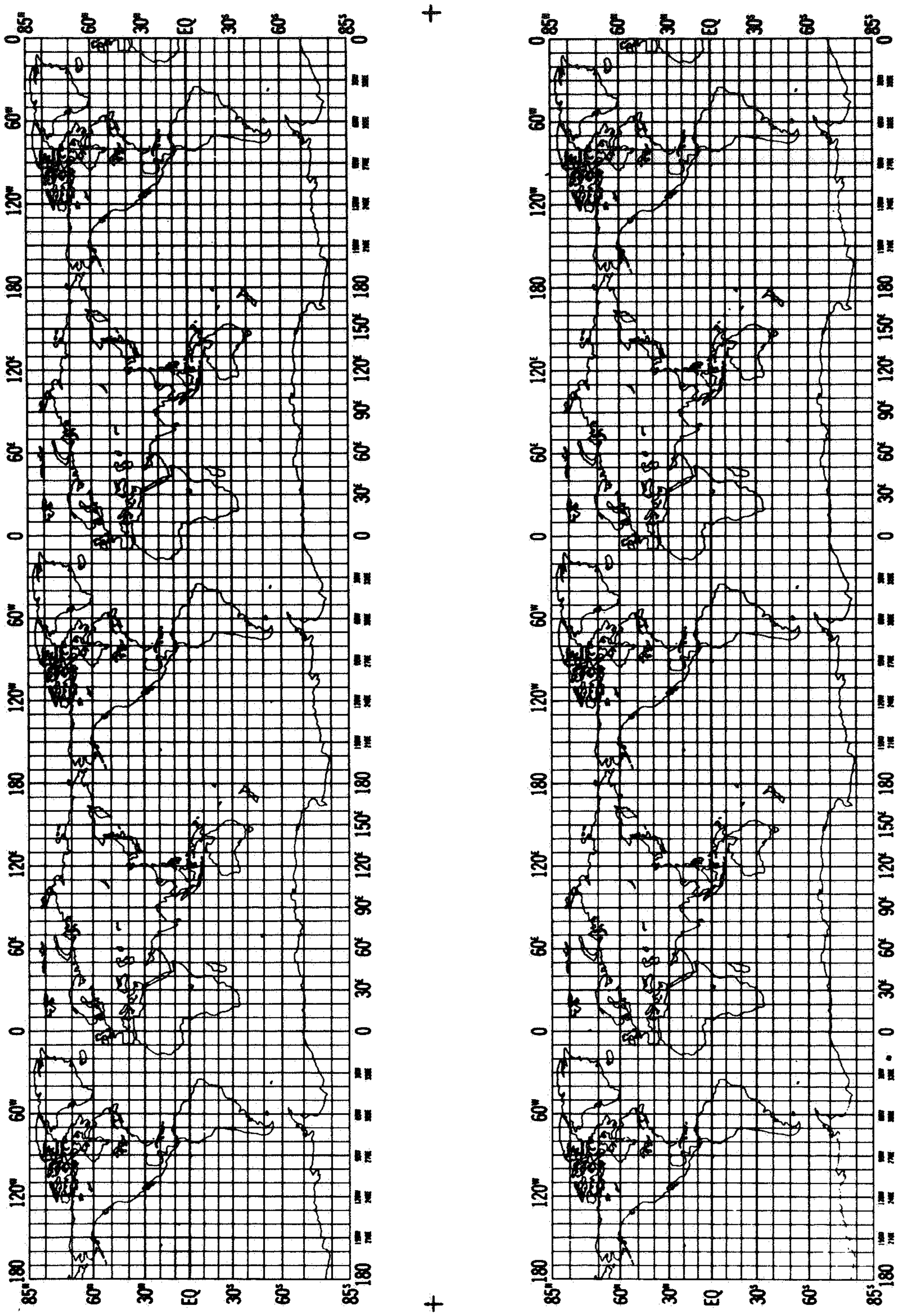
$$Y = 179.52 \log_{10} \text{Tan}\,(45^{o} + 0.4\,\varphi) + K$$

$$= 77.96 \ln \text{Tan}\,(45^{o} + 0.4\,\varphi) + K$$

where K = 707 for the upper strip and

K = 292 for the lower strip.

The area from Y = 0 to Y = 140 raster counts, from Y = 440 to Y = 560 and from Y = 860 to Y = 1023 is outside the map area and may be used for titles, etc.



(This Figure has been Rotated 90^{O} to the Left for Printing Purposes Only)

Figure E-5

APPENDIX F: MISCELLANEOUS INFORMATION

APPENDIX F

MISCELLANEOUS INFORMATION

This appendix has been provided as a potpourri of helpful suggestions for new and experienced S-C 4020 Plotter programmers. Certainly, as experience is gained in plotter programming, techniques are developed and particular peculiarities with regard to software packages or the plotter itself are uncovered. In order that all at the GSFC may benefit from an individual's experience and more effective use of the plotter may be obtained, such experienced findings should be jotted down and forwarded to the Programming Methods Section, located in Building No. 3, Room 125 at Goddard. Programming Methods may then review the suggestions and either document them for inclusion within this appendix or take some other corrective action.

1. (Hardware) The "00" is the S-C 4020 code for a carriage return. Therefore, any variable printed under an A format will produce up to 6 carriage returns if the cell is not filled (remaining all zeros from load time). This occurrence will cause the next legitimate character to be flushed left and down as many as six lines from where expected.

2. (Lockheed package) Inorder to decrease processing time when using the Lockheed package, the buffer areas may be increased from the normal size of 150 locations to 600 locations by the following code:

```
 △          ASM        BUFRV$, BUFRV$
$(0)
BUFRV$*     +          BUFAV, BUFBV
            +          0, 600
BUFAV       RES        600
BUFBV       RES        600
END
```